

Shifting Innovation to Users via Toolkits

Eric von Hippel • Ralph Katz

Sloan School of Management, Massachusetts Institute of Technology,

Cambridge, Massachusetts 02139

304 Hayden Hall, Northeastern University,

Boston, Massachusetts 02115

evhippel@mit.edu • r.katz@neu.edu

In the traditional new product development process, manufacturers first explore user needs and then develop responsive products. Developing an accurate understanding of a user need is not simple or fast or cheap, however. As a result, the traditional approach is coming under increasing strain as user needs change more rapidly, and as firms increasingly seek to serve “markets of one.”

Toolkits for user innovation is an emerging alternative approach in which manufacturers actually *abandon* the attempt to understand user needs in detail in favor of transferring *need-related* aspects of product and service development to users. Experience in fields where the toolkit approach has been pioneered show custom products being developed much more quickly and at a lower cost. In this paper we explore toolkits for user innovation and explain why and how they work.

(User Innovation; Toolkits; Mass Customization; Product Development)

1. Introduction

Research has consistently shown that new products and services must accurately respond to user needs if they are to succeed in the marketplace. However, it is often a *very* costly matter for firms to understand users' needs deeply and well. Need information is very complex, and conventional market research techniques only skim the surface. Techniques that probe more deeply, such as ethnographic studies, are both difficult and time consuming. Further, the task of understanding user needs is growing ever more difficult as firms increasingly strive to learn about and serve the unique needs of “markets of one,” and as the pace of change in markets and user needs grows ever faster. Indeed, firms at the leading edge of these trends are finding that conventional solutions are breaking down completely, and that a whole new approach is needed if they are to be able to continue to produce products and services that accurately respond to their users' needs.

Fortunately, an entirely new approach to this problem is being developed in a few high-tech fields. In this emerging new approach, manufacturers actually *abandon* their increasingly frustrating efforts to understand users' needs accurately and in detail. Instead, they outsource key *need-related* innovation tasks to the users themselves after equipping them with appropriate “toolkits for user innovation.”

Toolkits for user innovation are coordinated sets of “user-friendly” design tools that enable users to develop new product innovations for themselves. The toolkits are not general purpose. Rather, they are specific to the design challenges of a specific field or subfield, such as integrated circuit design or software product design. Within their fields of use, they give users real freedom to innovate, allowing them to develop producible custom products via iterative trial and error. That is, users can create a preliminary design, simulate or prototype it, evaluate its functioning in their own use environment, and then iteratively improve it until satisfied.

Toolkits for user innovation first emerged in a primitive form in the 1980s in the high-tech field of custom integrated circuit design and manufacturing. In this field, as IC products grew increasingly large and complex, the costs of not understanding user needs precisely and completely at the start of product design work had grown to punishingly high levels. Many errors as a result of incomplete or inaccurate specification of user needs were occurring, and the cost of correcting even a single error found late in the design process or during user testing could involve literally months of delay and hundreds of thousands of dollars of extra engineering charges. The introduction of the toolkits approach to the custom semiconductor field have reduced development time by two-thirds or more for products of equivalent complexity (von Hippel 1998). Semiconductor manufacturers' sales of user-designed chips were \$15 billion in 2000 (Thomke and von Hippel 2002).

Although toolkits for user innovation are now only applied to the development of a few types of custom industrial products and services, we propose that they will eventually be a valuable product development method for all product types characterized by heterogeneous user demand. As we will see in this paper, the economics of sticky information make toolkits desirable under many conditions, while technical advances in computerization are making them increasingly practical in many fields.

In this paper we begin by explaining the benefits of shifting need-related design activities to users (§2). We then explore how this can be achieved via "toolkits for user innovation" and detail the elements that such a toolkit should contain (§3). Finally, we discuss the relationship of toolkits for user innovation to other development methods, and where they can be most effectively applied (§4).

2. Toolkits and Sticky Information

The toolkits approach to product and service development involves transferring *need-related* product development tasks from manufacturers to users, and equipping the users with tools to carry out those tasks. To understand the utility of such a transfer consider that, to solve a problem, needed information and

problem-solving capabilities (also a form of information) must be brought together at a single locus. The requirement to transfer information from its point of origin to a specified problem-solving site will not affect the locus of problem-solving activity when that information can be shifted at little or no cost. However, when it is costly to transfer from one site to another in useable form—it is, in our terms, sticky—the distribution of problem-solving activities can be significantly affected.

The stickiness of a given unit of information in a given instance is defined as the incremental expenditure required to transfer it to a specified locus in a form useable by a given information seeker. When this cost is low, information stickiness is low; when it is high, stickiness is high (von Hippel 1994). A number of researchers have both argued and shown that information required by technical problem solvers is indeed often costly to transfer for a range of reasons. Information stickiness can be due to attributes of the information itself, such as the way it is encoded (Nelson 1982, 1990; Pavitt 1987; Rosenberg 1982). In addition, it can be due to attributes of information seekers or providers. For example, a particular information seeker may be less able in acquiring information because of a lack of certain tools or complementary information—a lack of "absorptive capacity" in the terminology of Cohen and Levinthal (1990). Also, specialized personnel such as "technological gatekeepers" (Allen 1984, Tushman and Katz 1980, Katz 1997) and specialized organizational structures such as information transfer groups (Katz and Allen 1988) can significantly affect information transfer costs between and within organizations.

In the case of product development, sticky information needed by developers is generated at both product manufacturer and product user sites. Generally a manufacturer has information regarding solution possibilities and its production process, while users have information about needs and the setting of use. The toolkits approach to product and service development reduces sticky information transfer costs by repartitioning the overall product development task into subtasks, each primarily requiring information from either the user or the manufacturer site. Then, it

assigns each subtask to user- or manufacturer-based problem solvers as appropriate.

Repartitioning of innovation process tasks for this purpose can involve fundamental changes to the underlying architecture of a product or service. Consider, for example, how semiconductor manufacturers shifted to the new toolkits paradigm for custom chip development. Traditionally, manufacturers of custom semiconductors had carried out all chip design tasks themselves, guided only by need specifications from users. And, because manufacturer development engineers were carrying out all design tasks, those engineers had typically incorporated need-related information into the design of both the fundamental elements of a circuit, such as transistors, and the electrical “wiring” that interconnected those elements into a functioning circuit.

The brilliant insight that allowed custom integrated circuit design to be partitioned into solution- and need-related subtasks was that the design of the chip’s fundamental elements, such as its transistors, could be made standard for all custom digital circuit designs. This subtask required rich access to the manufacturer’s sticky solution information regarding how semiconductors are fabricated, but did not require detailed information on specific user needs. It could therefore be assigned to manufacturer-based chip design and fabrication engineers. It was also observed that the subtask of interconnecting standard digital circuit elements into a functioning integrated circuit required only sticky, need-related information about chip function—for example, whether it was to function as a microprocessor for a calculator or the voice chip for a robotic dog. This subtask of “wiring” the circuit was therefore assigned to users—the parties already in possession of the relevant need-related information. In other words, this new type of chip, called a “gate array,” had a novel architecture created specifically to separate problem-solving tasks requiring access to a manufacturer’s sticky solution information from those requiring access to users’ sticky need information. Tasks involving sticky solution information were then assigned to chip manufacturers, while those involving sticky need information were assigned to users.

The same basic principle can be illustrated in a less technical context—food design. In this field, manufacturer-based designers have traditionally undertaken the entire job of developing a novel food, and so they have freely blended need-specific design into any or all of the recipe-design elements wherever convenient. For example, manufacturer-based developers might find it convenient to create a novel cake by both designing a novel flavor and texture for the cake body, and designing a complementary novel flavor and texture into the frosting. However, it is possible to repartition these same tasks so that only a few draw upon need-related information, and these can then be more easily transferred to users.

The architecture of the humble pizza illustrates how this can be done. In the case of the pizza, many aspects of the design, such as the design of the dough and the sauce, have been made standard, and user choice has been restricted to a single task—design of toppings. In other words, all need-related information that is unique to a given user has been linked to the toppings-design task only. Transfer of this single design task to users can still potentially offer creative individuals a very large design space to play in (although pizza shops typically restrict it sharply). Any edible ingredients one can think of—from eye of newt to edible flowers—are potential topping components, but the fact that need-related information has been concentrated within only a single product design task makes it much easier to transfer design freedom to the user.

Once problem solving and the sticky information needed to perform it have been colocated, the development of new products and services can proceed much more rapidly and effectively. To understand why this is so, consider that problem solving in general, and development of a new product or service in particular, proceeds via an iterative process of trial and error (Barron 1988, von Hippel and Tyre 1995). User- or manufacturer-based designers begin by designing what they think they want; then they test the initial solution, find drawbacks, and try again. This iterative process is sometimes called “learning by doing” (Arrow 1962, Rosenberg 1982). When tasks have been subdivided so that the sticky information required to solve them and the problem solvers are

colocated, the need to shift problem solving back and forth *between* user and manufacturer during the trial and error cycles involved in learning by doing is eliminated. Iterative learning by doing is still carried out, but the trial-and-error cycles for each subtask are carried out entirely *within* a user or manufacturer firm.

To appreciate the major advantage in problem-solving speed and efficiency that concentrating problem solving within a single locus can create, consider a familiar everyday example: The contrast between conducting financial strategy development with and without “user-operated” financial spreadsheet software.

- Prior to the development of easy-to-use financial spreadsheet programs such as Lotus 1-2-3 and Microsoft’s Excel, a CFO might have carried out a financial strategy development exercise as follows. First, the CFO would have asked his or her assistant to develop an analysis incorporating a list of assumptions. A few hours or days might elapse before the result was delivered. Then the CFO would use her rich understanding of the firm and its goals to study the analysis. She would typically almost immediately spot some implications of the patterns developed, and would then ask for additional analyses to explore these implications. The assistant would take the new instructions and go back to work while the CFO switched to another task. When the assistant returned, the cycle would repeat until a satisfactory outcome was found.

- After the development of financial spreadsheet programs, a CFO might begin an analysis by asking an assistant to load up a spreadsheet with corporate data. The CFO would then “play with” the data, trying out various ideas and possibilities and “what if” scenarios. The cycle time between trials would be reduced from days or hours to minutes. The CFO’s full, rich information would be applied immediately to the effects of each trial. Unexpected patterns—suggestive to the CFO but often meaningless to a less knowledgeable assistant—would immediately be identified and followed up, and so forth.

It is generally acknowledged that spreadsheet software that enables expert users to “do it themselves” has led to better outcomes that are achieved faster

(Levy 1984, Schrage 2000). The advantages are similar in the case of product and service development. Thus, when custom integrated circuit design is carried out entirely by manufacturers, users cannot engage in learning by doing with respect to their need and their use environment until a chip has been completely designed by the manufacturer and sample chips have been made available. At that late stage, as was noted earlier, it can cost months and hundreds of thousands of dollars for a manufacturer to incorporate modifications requested by users based upon learning by doing. In contrast, users can learn to identify and correct need-related design errors early, rapidly, and at a very low cost if they are equipped with an appropriate toolkit for user innovation. Learning by doing via trial and error still occurs, of course, but the cycle time is much faster because the complete cycle of need-related learning is carried out at a single user site earlier in the development process.

3. Toolkits—A Way to Transfer Design Capability to Users

In principle, then, when “need-related” design tasks are assigned to users and “solution-related” tasks are assigned to manufacturers, times and costs are compressed, and learning by doing is more effectively integrated into the design process. However, users are not design specialists in the manufacturer’s product or service field, so how can one expect them to create sophisticated, producible custom designs efficiently and effectively? Manufacturers who have pioneered in this field have solved the problem by providing users with kits of design tools that can help them to carry out the design tasks assigned to them (von Hippel 1998).

Toolkit development involves “unsticking” manufacturer solution and production information relevant to the development work of user-innovators and incorporating it into a toolkit. This can be done because the stickiness of a given unit of information is not immutable. Rather, it can be reduced by investments made to that end. For example, firms may reduce the stickiness of a critical form of technical expertise by investing in converting some of that expertise from tacit knowledge to the more explicit

and easily transferable form of a software “expert system” (Davis 1986). Also, they may invest in reducing the stickiness of information of interest to a particular group of users by encoding it in the form of a remotely accessible computer data base. This is what the travel industry did, for example, when it invested substantial sums to put its various data bases for airline schedules, hotel reservations, and car rentals “online” in a user-accessible form.

The incentive to invest in reducing the stickiness of a given unit of information will vary *according to the number of times that one expects to transfer it*. As illustration, suppose that to solve a particular problem, two units of equally sticky local information are required, one from a user and one from a manufacturer. In that case, there will be an equal incentive operating to unstick either of these units of information to reduce the cost of transfer, other things (such as the cost of unsticking) being equal. But now suppose that there is reason to expect that one of the units of information, say the manufacturer’s, will be a candidate for transfer n times in the future, while the user’s unit of information will be of interest to problem solvers only once. For example, suppose that a manufacturer expects to have the same technical information called on repeatedly to solve n user product application problems, and that each such problem involves unique user information. In that case, the total incentive to unstick the manufacturer’s information across the entire series of user problems is n times higher than the incentive for an individual user to unstick its problem-related information.

In the case of the problem-solving work of product and service development, the situation just described is the one encountered when user needs for a given product type are heterogeneous. Under these conditions, manufacturers specializing in a given product type attempt to adapt the same basic approach to the diverse application problems of many users. For example, manufacturers of adhesives will attempt to solve diverse user fastening-related problems with specialized adhesives, while manufacturers of mechanical fasteners will attempt to solve such problems with specialized screws and bolts. The commonality in solution approach means that the sticky information required from a manufacturer to solve

each novel application problem tends to be the same, involving such things as the properties and limitations of the solution type. In contrast, the diversity in applications means that sticky information required from users tends to be novel or have novel components. Thus, the higher the heterogeneity of user needs faced by a manufacturer, the higher its incentive to invest in unsticking problem-related information relevant to user problem solvers and transfer that information to users in the form of a toolkit for user innovation (von Hippel 1998).

3.1. Elements of a Toolkit

Toolkits for innovation are not new as a general concept—every manufacturer equips its engineers with a set of tools suitable for designing the type of products or services it wishes to produce. Toolkits for users also are not new—many users have personal toolsets that they have assembled to help them create new items or modify standard ones. For example, some users have woodworking tools ranging from saws to glue, which can be used to create or repair furniture. Others may have software tools to write or modify software. What is new, however, is integrated toolsets to enable users to create and test designs for custom products or services that can then be produced “as is” by manufacturers.

We propose that effective toolkits for user innovation will enable five important objectives. First, they will enable users to carry out complete cycles of trial-and-error learning. Second, they will offer users a “solution space” that encompasses the designs they want to create. Third, users will be able to operate them with their customary design language and skills—in other words, well-designed toolkits are “user friendly” in the sense that users do not need to engage in much additional training to use them competently. Fourth, they will contain libraries of commonly used modules that the user can incorporate into his or her custom design, thus allowing the user to focus his or her design efforts on the truly unique elements of that design. Fifth and finally, properly designed toolkits will ensure that custom products and services designed by users will be producible on manufacturer production equipment *without* requiring revisions by manufacturer-based engineers.

Learning by Doing via Trial-and-Error. As was mentioned earlier, it is important that toolkits for user innovation enable users to go through complete trial-and-error cycles as they create their designs. Such cycles begin with the design of a possible solution. The solution is then built (or simulated on a computer), tested, and evaluated. If evaluation shows that improvements are needed, the cycle is repeated. For example, suppose that a user is designing a new custom telephone-answering system for her firm, using a software-based CTI design toolkit provided by a vendor. Suppose also that the user decides to include a new rule to “route all calls of X nature to Joe” in her design. A properly designed toolkit would allow her to temporarily place the new rule into the telephone system software so that she could actually try it out (via a real test or a simulation) and see what happened. She might discover that the solution worked perfectly. Or, she might find that the new rule caused some unexpected form of trouble—for example, Joe might be flooded with too many calls—in which case it would be “back to the drawing board” for another design and another trial.

In the same way, toolkits for user innovation in the semiconductor design field allow the users to design a circuit that they think will meet their needs and then test the design by “running” it in the form of a computer simulation. This quickly reveals errors that the user can then quickly and cheaply fix using toolkit-supplied diagnostic and design tools (Thomke 1998). For example, a user might discover by testing a simulated circuit design that he or she had forgotten about a switch to adjust the circuit—and make that discovery simply by trying to make a needed adjustment. The user could then quickly and cheaply design in the needed switch without major cost or delay.

One can appreciate the importance of giving the user the capability for trial-and-error learning by doing in a toolkit by thinking about the consequences of not having it. When users are not supplied with toolkits that enable them to draw on their local, sticky information and engage in trial-and-error learning, they must actually order a product and have it built to learn about design errors—typically a very costly and unsatisfactory way to proceed. For example, custom furniture makers allow customers to select from

a range of options for their furniture, but they do not offer the customer a way to learn during the design process and before buying. The cost to the customer is unexpected learning that comes too late: “That style of couch and swatch of fabric did look great in the showroom. But now that the couch has been delivered, I discover that it makes the room feel crowded, and that the color of the fabric clashes with the wallpaper!”

An Appropriate “Solution Space.” Economical production of custom products and services is only achievable when a custom design falls within the pre-existing capability and degrees of freedom built into a given manufacturer’s production system. We may term this the “solution space” offered by that system. A solution space may vary from very large to small, and if the output of a toolkit is tied to a particular production system, the design freedom that a toolkit can offer a user will be accordingly large or small. For example, the solution space offered by the production process of a custom integrated circuit manufacturer offers a huge solution space to users—it will produce any combination of logic elements interconnected in any way that a user-designer might desire, with the result that the user can invent anything from a novel type of computer processor to a novel “silicon organism” within that space. However, note that the semiconductor production process also has stringent limits. It will only implement product designs expressed in terms of semiconductor logic—it will not implement designs for bicycles or houses. Also, even within the arena of semiconductors, it will only be able to produce semiconductors that fit within a certain range with respect to size and other properties. Another example of a production system offering a very large solution space to designers—and, potentially to user-designers via toolkits—is the automated machining center. Such a device can basically fashion any shape out of any machinable material that can be created by any combination of basic machining operations such as drilling and milling. As a consequence, toolkits for user innovation intended to create designs

producable on automated machining centers can offer users access to that very large solution space.¹

Large solution spaces can typically be made available to user-designers when production systems and associated toolkits allow users to manipulate and combine relatively basic and general-purpose building blocks and operations, as in the examples above. In contrast, small solution spaces typically result when users are only allowed to combine a relatively few special-purpose “options.” Thus, users who want to design their own custom automobile are restricted to a relatively small solution space: They can only make choices from lists of options regarding such things as engines, transmissions, and paint colors. Similarly, purchasers of eyeglasses produced by “mass-customization”² production methods are restricted to combining “any frame from this list” of pre-designed frames with “any hinge from that list” of pre-designed hinges, and so on.

The reason producers of custom products or services enforce constraints on the solution space that user-designers may use is that custom products can only be produced at reasonable prices when custom user designs can be implemented by simply making low-cost adjustments to the production process. This condition is met within the solution space on offer. However, responding to requests that fall outside of that space will require small or large additional investments by the manufacturer. For example, an integrated circuit producer may have to invest many millions of dollars and rework an entire production process to respond to a customer request for a larger chip that falls outside of the solution space associated with its present production equipment.

“User-Friendly” Toolkits. Toolkits for user innovation are most effective and successful when they

are made “user friendly” by enabling users to use the skills they already have and work in their own customary and well-practiced design language. This means that users don’t have to learn the—typically different—design skills and language customarily used by manufacturer-based designers, and so will require much less training to use the toolkit effectively.

For example, in the case of custom integrated circuit design, toolkit users are typically electrical engineers who are designing electronic systems that will incorporate custom ICs. The digital IC design language normally used by electrical engineers is Boolean algebra. Therefore, user-friendly toolkits for custom IC design are provided that allow toolkit users to design in this language. That is, users can create a design, test how it works, and make improvements all within their own customary language. At the conclusion of the design process, the toolkit then translates the user’s logical design into a different form, the design inputs required by the IC manufacturer’s semiconductor production system.

A design toolkit based on a language and skills and tools familiar to the user is only possible, of course, to the extent that the user *has* familiarity with some appropriate and reasonably complete language and set of skills and tools. Interestingly, this is the case more frequently than one might initially suppose, at least in terms of the *function* that a user wants a product or service to perform—because functionality is a face that the product or service presents to the user. (Indeed, an expert user of a product or service may be much more familiar with that functional “face” than manufacturer-based experts.)

Thus, the user of a custom semiconductor is the expert in what he or she wants that custom chip to *do*, and is skilled at making complex trade-offs among familiar functional elements to achieve a desired end. Thus: “If I increase chip clock speed, I can reduce the size of my cache memory and . . .” As a less technical example, consider the matter of designing a custom hair style. In this field there is certainly a great deal of information known to hairstylists that even an expert user may not know, such as how to achieve a given look via “layer cutting,” or how to achieve a given streaked color pattern by selectively dyeing

¹ Note, however, that current computer-aided design and manufacturing software (CAD–CAM) is not equivalent to a toolkit for user innovation. It does not, for example, offer users the ability to conduct trial-and-error tests of the functional suitability of the designs they are constructing.

² “Mass-customized” production systems are systems of computerized process equipment that can be adjusted instantly and at low cost. Such equipment can produce small volumes of a product or even one-of-a-kind products at near mass-production costs (Pine 1993).

some strands of hair. However, an expert user is often very well practiced at the skill of examining the shape of his or her face and hairstyle as reflected in a mirror, and visualizing specific improvements that might be desirable in matters such as curls or shape or color. In addition, the user will be very familiar with the nature and functioning of everyday tools used to shape hair, such as scissors and combs.

A “user-friendly” toolkit for hairstyling innovation can be built upon these familiar skills and tools. For example, a user can be invited to sit in front of a computer monitor and study an image of his or her face and hairstyle as captured by a video camera. Then, she can select from a palette of colors and color patterns offered on the screen, can superimpose the effect on her existing hairstyle, can examine it, and repeatedly modify it in a process of trial-and-error learning. Similarly, the user can select and manipulate images of familiar tools such as combs and scissors to alter the image of the length and shape of her own hairstyle as projected on the computer screen, can study and further modify the result achieved, and so forth. Note that the user’s new design can be as radically new as desired, because the toolkit gives the user access to the most basic hairstyling variables and tools, such as color and scissors. When the user is satisfied, the completed design can be translated into technical hairstyling instructions in the language of a hairstyling specialist—the intended “production system” in this instance.

Module Libraries. Custom designs are seldom novel in all their parts. Therefore, libraries of standard modules that will frequently be useful elements in custom designs are a valuable part of a toolkit for user innovation. Provision of such standard modules enables users to focus their creative work on those aspects of their design that are truly novel. Thus, a team of architects who are designing a custom office building will find it very useful to have access to a library of standard components, such as a range of standard structural support columns with preanalyzed structural characteristics, that they can incorporate into their novel building designs. Similarly, designers of custom integrated circuits find it very useful to incorporate predesigned elements in their custom designs, ranging from simple

operational amplifiers to complete microprocessors—examples of “cells” and “macrocells,” respectively—that they draw from a library in their design toolkit. And again similarly, even users who want to design quite unusual hairstyles will often find it helpful to begin by selecting a hairstyle from a toolkit library. The goal is to select a style that has some elements of the desired look. Users can then proceed to develop their own desired style by adding to and subtracting from that starting point.

Translating User Designs for Production. Finally, the “language” of a toolkit for user innovation must be convertible without error into the “language” of the intended production system at the conclusion of the user design work. If this is not so, then the entire purpose of the toolkit is lost, because a manufacturer receiving a user design essentially has to “do the design over again.” Error-free translation need not emerge as a major problem—for example, it was never a major problem during the development of toolkits for integrated circuit design, because both chip designers and integrated circuit component producers already used a language based on digital logic. On the other hand, in some fields, translating from the design language preferred by users to the language required by intended production systems can be *the* problem in toolkit design. To illustrate, consider the case of a recent Nestle USA’s FoodServices Division toolkit test project developed for use in custom food design by the Director of Food Product Development, Ernie Gum.

One major business of Nestle FoodServices is production of custom food products such as custom Mexican sauces for major restaurant and takeout food chains. Custom foods of this type have been traditionally developed by or modified by chain executive chefs, using what are in effect design and production toolkits taught by culinary schools: Restaurant-style recipe development based on food ingredients available to individuals and restaurants, processed on restaurant-style equipment. After using their traditional toolkits to develop or modify a recipe for a new menu item, executive chefs call in Nestle Foodservices or other custom food producers and ask them to manufacture the product they have designed—and

this is where the language translation problem rears its head.

There is no error-free way to “translate” a recipe expressed in the “language” of a traditional restaurant-style culinary toolkit into the “language” required by a food manufacturing facility. Food factories can only use ingredients that are obtainable in quantity at a consistent quality. These are not the same as and may not taste quite the same as ingredients used by the executive chef during recipe development. Also, food factories use volume production equipment, such as huge steam-heated retorts. Such equipment is very different from restaurant-style stoves and pots and pans, and it often cannot reproduce the cooking conditions created by the executive chef on his stovetop—for example, very rapid heating. Therefore, food production factories cannot simply produce a recipe developed by or modified by an executive chef “as is” under factory conditions—it will not taste the same.

As a consequence, even though an executive chef creates a prototype product using a traditional chef’s toolkit, food manufacturers find most of that information—the information about ingredients and processing conditions—useless because it cannot be straightforwardly translated into factory-relevant terms. The only information that can be salvaged is the information about taste and texture contained in the prototype. Therefore, production chefs carefully examine and taste the customer’s custom food prototype, and then try to make something that “tastes the same” using factory ingredients and methods. However, executive chef taste buds are not necessarily the same as production chef taste buds, and so the initial factory version—and the second and the third—is typically not what the customer wants. As a result, the producer must create variation after variation until the customer is finally satisfied. In the case of Nestle, this painstaking “translation” effort means that it often takes 26 weeks to bring a new custom food product from chef’s prototype to first factory production.

To solve the translation problem, Gum created a novel toolkit of food “precomponent” ingredients to be used by executive chefs during food development. Each ingredient in the toolkit is the Nestle *factory* version of an ingredient traditionally used by chefs

during recipe development: That is, it is an ingredient commercially available to Nestle that had been processed as an independent ingredient on Nestle factory equipment. For example, a toolkit designed for Mexican chefs (the first one designed by Nestle) contains a chili puree ingredient processed on industrial equipment identical to that used to produce food in commercial-sized lots. Each precomponent also contains traces of materials that will interact during production—for example, traces of a tomato “carrier” are included in the chili puree—so that the taste effects of such interactions are also included in the precomponent.

Chefs using the toolkit of Nestle precomponents to develop new product prototypes do find that each component differs slightly from the fresh components he or she is used to. However, these differences are discovered immediately via “learning by doing,” and the chef then immediately adapts and moves to the desired final taste and texture by making trial-and-error adjustments in the ingredients and proportions in the recipe being developed. When a recipe based on precomponents is finished, it can be immediately and precisely reproduced by Nestle factories—because now the user-developer is using the same language as the factory for his or her design work. In the Nestle case, testing shows that adding the “error-free translation” feature to toolkit-based design by users can potentially shorten the time of custom food development from 26 weeks to 3 weeks by eliminating repeated redesign and refinement interactions between Nestle and its custom food customers.

4. Discussion

To this point we have explored why toolkits for user innovation can be valuable, and have developed the contents of a toolkit. We now conclude by discussing the relationship of toolkits to other product development methods, where toolkits will offer the most value, how toolkits can be developed, and the competitive value of toolkits for manufacturers.

Relationship to Other Product Development Methods. Toolkits for user innovation improve the ability of users to innovate for themselves. Users with

sufficient incentive to do so can apply toolkits to design products and services that fit their own needs precisely, at a lower cost than would otherwise be the case. "Product configurators" used by producers of mass-customized products are similar in intent but less capable than toolkits. They invite product purchasers to configure their own unique product by selecting from lists of options that have been pre-designed by the mass customizer. For example, Dell Computer invites visitors to its website to "design your own computer" by making choices among lists of computer components on offer, such as monitors and disk drives.

Market research techniques conventionally used for product design such as multiattribute techniques and conjoint analysis have a very different basic purpose. These are used to collect and analyze need and preference information from many individual users. The information is used by manufacturer-based product developers to design standard products that will bring the greatest satisfaction to the greatest number of customers. Products are not designed by users themselves.

Lead user idea-generation methods are similar to conventional market research methods in purpose, but allocate idea generation to lead users rather than to in-house developers. Thus, lead user studies begin with market and trend analyses to determine the nature and direction of migration of user preference. Then, potential design solutions are sought from lead users located at the leading edge of important market trends identified. The goal is to incorporate one or a few of these solutions into standard products that will address the preferences of as many as users as possible to the greatest extent possible (von Hippel et al. 1999).

The toolkit for innovation approach is complementary to the lead user approach in an interesting way. Some of the users choosing to employ a toolkit to design a product precisely right for their own needs will be "lead users," whose present strong need foreshadows a general need in the marketplace. Manufacturers can find it valuable to identify and acquire the generally useful improvements made by these lead users, and then supply them to the general market. The business model of Stata Corporation illustrates

this pattern. Stata Corporation sells a software package for performing complex statistical analyses. The package offers the functions of a toolkit for user innovation, and Stata encourages its customers to create and share new software code for executing novel statistical techniques. The company then selects user developments of interest to many users and adapts and incorporates these into its next product release.

Where Toolkits Offer the Most Value. Toolkits for user innovation are applicable to essentially all types of products and services where heterogeneity of user demand makes custom, "precisely right" solutions valuable to buyers. As market researchers have long known, many markets have high heterogeneity of demand (Franke and Reisinger 2002). The toolkits for the user innovation approach are becoming more attractive in such fields as advances in both computerized design and computerized production technologies progressively reduce the fixed costs associated with the design and production of novel products.

The fixed costs of design are being steadily reduced by the refinement and increased application of computer-aided design (CAD) tools. These design tools have sharply reduced the costs of designing a unique product for product producers. When they are simplified and transferred to users in the form of "user-friendly" toolkits described in this paper, they do the same for users. The fixed costs of tooling have been sharply reduced by the introduction of "mass-customized" production methods. These methods involve various combinations of computerized production machines that can be adjusted to produce different outputs nearly instantly and with low-cost modular product design and flexible assembly techniques. Manufacturers using mass-customized production can often make even single-unit quantities of custom products at a cost that is reasonably competitive with the costs of manufacturing similar items by traditional mass-production methods (Pine 1993).

We should note that toolkits are not the appropriate solution for all product development needs, even in highly heterogeneous markets. They do allow greater scope for users to apply their understanding of a need more directly, and thus will generally result in products that "fit the need" better. On the other hand,

toolkits will not be the preferred approach when the highest achievable performance on other dimensions is required, because they incorporate automated design rules that cannot, at least at present, translate designs into products or software with the same skill as can a human designer. For example, a design for a gate array generated via toolkit will typically take up more physical space on a silicon chip than would a full-custom design of similar complexity. Even when toolkits are on offer, therefore, manufacturers may continue to design certain products (those with difficult technical demands) while customers take over the design of others (those involving complex or rapidly evolving user needs).

We should also note that the design freedom provided by toolkits for user innovation may not be of interest to all or even to most users in a market characterized by heterogeneous needs. A user must have a need for something different that is strong enough to offset the costs of putting a toolkit to use. Toolkits may therefore be offered only to the subset of users who have a need for them. Or, in the case of software, toolkits may be provided to all users along with a standard "default" version of the product or service, because the cost of delivering the extra software is essentially zero. In such a case the toolkit capability will simply lie unused in the background unless and until a user has sufficient incentive to evoke and employ it.

Development of Toolkits. We have said that manufacturers that offer toolkits for user innovation to their customers are freed from having to know the details of their customers' needs for new products and services. On the other hand, the manufacturer does still have to know the solution space his customers require to be able to design the novel products or services they want. For example, Nestle has to know which 30 ingredients to put into its Mexican sauce design toolkit, even if it does *not* have to know anything about a specific customer's need, or anything about the attributes of the sauce that customer hopes to make.

Fortunately, determining the solution dimensions that a toolkit must offer does not take superhuman insight on the part of manufacturer experts.

Manufacturer-based developers can create a first-generation toolkit by analyzing existing customer products and determining the dimensions that were required to design those. Alternatively, manufacturers can simply modify existing in-house design toolsets to make them more user friendly, and distribute these as a first-generation toolkit for user innovation. All that is required for initial success is that a first-generation toolkit offer enough functionality to make it valuable to interested users relative to other existing options. As users begin to apply the toolkit to their projects, the more advanced among them will "bump up against the edges" of the solution space on offer and then request the additional capabilities they need to implement their novel designs. Manufacturers can then improve their toolkits by responding to these explicit requests for improvement, or they can wait until impatient lead users actually create and test and use the toolkit improvements they need for themselves. Toolkit improvements that prove to be of general value can then be incorporated into the standard toolkit and distributed to the general toolkit-using community just as product improvements developed by lead users can be distributed to the general community of users.

Competitive Value of Toolkits for Manufacturers.

Toolkits can create competitive advantages for manufacturers that are first to offer them. Being first into a marketplace with a toolkit may yield first-mover advantages with respect to setting a standard for a user design language that has a good chance of being generally adopted by the user community in that marketplace. Also, manufacturers may tailor the toolkits they offer to allow easy, error-free translations of designs made by users into their own production capabilities. This gives originators a competitive edge even if the toolkit language itself becomes an open standard. For example, in the field of custom food production, customers often try to get a better price by asking a number of firms to quote on producing the prototype product they have designed. If a design has been created on a toolkit based on a Nestle-developed language of precomponents that can be produced efficiently on Nestle factory equipment by methods known best to that firm—Nestle

will obviously enter the contest with a competitive edge.

Toolkits can impact existing business models in a field in ways that may or may not be to manufacturers' competitive advantage in the longer run. For example, consider that many manufacturers of products and services appropriate benefit from both their design capabilities and their production capabilities. A switch to user-based customization via toolkits can affect their ability to do this over the long term. Thus, a manufacturer that is early in introducing a toolkit approach to custom product or service design may initially gain an advantage by tying that toolkit to his particular production facility. However, when toolsets are made available to customer designers, this tie often weakens over time. Customers and independent tool developers can eventually learn to design toolkits applicable to the processes of several manufacturers. (Indeed, this is precisely what has happened in the custom integrated circuit industry. The initial toolsets revealed to users by producers of custom integrated circuits were producer specific. Over time however, specialist tool design firms such as Cadence developed toolkits that enabled users to make designs producible by a number of vendors.) The end result is that manufacturers that previously benefited from selling their product design skills and production skills can eventually be forced by the shifting of design tasks to customers via toolkits to a position of benefiting from production skills only.

However, manufacturers who project long-term disadvantages that may accrue from a switch to a toolkit-based innovation process will not necessarily have the luxury of declining to introduce one. If any manufacturer introduces the toolkits approach into a field favoring its use, customers will tend to migrate to it, forcing competitors to follow. Therefore, a firm's only real choice in a field where conditions are favorable to the introduction of toolkits is the choice of leading or following.

We conclude by proposing, as we did at the start of this article, that toolkits for user innovation will eventually be adopted by many manufacturers facing heterogeneous customer demand. As toolkits are more generally adopted, the organization of innovation-related tasks seen today especially in the field of custom integrated circuit production will spread, and

users will increasingly be able to get *exactly* the products and services they want—by designing them for themselves.

References

- Allen, T. 1984. *Managing the Flow of Technology*. MIT Press, Cambridge, MA.
- Arrow, Kenneth J. 1962. Economic welfare and the allocation of resource of invention. Richard R. Nelson, ed. *The Rate and Direction of Inventive Activity: Economic and Social Factors*. A Report of the National Bureau of Economic Research, Princeton Universal Press, Princeton, NJ, 609–625.
- Baron, Jonathan. 1988. *Thinking and Deciding*. Cambridge University Press, New York.
- Cohen, Wesley M., Daniel A. Levinthal. 1990. Absorptive capacity: A new perspective on learning and innovation. *Admin. Sci. Quart.* 35(1) 128–152.
- Davis, Randall. 1986. Knowledge-based systems. *Science* 231(4741) 957–963.
- Franke, Nik, H. Reisinger. 2002. Remaining within-cluster variance in cluster analyses: A meta-analysis. Working paper, Vienna University of Economics and Business Administration, Vienna, Austria.
- Katz, Ralph. 1997. *The Human Side of Managing Technological Innovation*. Chapters 18 and 35, Oxford University Press, New York.
- , Thomas J. Allen. 1988. Organizational issues in the introduction of new technologies. Ralph Katz, ed. *Managing Professionals in Innovative Organizations*, Ballinger, Cambridge, MA, 442–456.
- Levy, Steven. 1984. A spreadsheet way of knowledge. *Harper's* 269(1614) 58–64.
- Nelson, Richard R. 1982. The role of knowledge in R&D efficiency. *Quart. J. Econom.* 97(3) 453–470.
- . 1990. What is public and what is private about technology? Consortium on competitiveness and cooperation, Working paper no. 90-9, Center for Research in Management, University of California at Berkeley, Berkeley, CA.
- Ogawa, Susumu. 1998. Does sticky information affect the locus of innovation? Evidence from the Japanese convenience-store industry. *Res. Policy* 26 777–790.
- Pavitt, Keith. 1987. The objectives of technology policy. *Sci. Pub. Policy*, 14(4) 182–188.
- Pine, Joseph B., II 1993. *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press, Cambridge, MA.
- Rosenberg, Nathan. 1982. *Inside the Black Box: Technology and Economics*. Cambridge University Press, New York. 131.
- Schrage, Michael. 2000. *Serious Play: How The World's Best Companies Simulate to Innovate*, Harvard Business School Press, Cambridge, MA. 38.
- Thomke, Stefan. 1998. Managing experimentation in the design of new products. *Management Sci.*, 44(6) 743–762.

- , Eric von Hippel. 2002. Customers as innovators: A new way to create value. *Harvard Bus. Rev.* **80**(4) 74–81.
- Tushman, Michael, Ralph Katz. 1980. External communication and project performance: An investigation into the role of gatekeepers. *Management Sci.* **26**(11) 1071–1085.
- von Hippel, Eric. 1994. Sticky information and the locus of problem solving: Implications for innovation. *Management Sci.* **40**(4) 429–439.
- . 1998. Economics of product development by users: The impact of “sticky” local information. *Management Sci.* **44**(5) 629–644.
- , Marcie Tyre. 1995. How “learning by doing” is done: Problem identification in novel process equipment. *Res. Policy* **24** 1–12.
- , Stephan Thomke, Mary Sonnack. 1999. Creating breakthroughs at 3M. *Harvard Bus. Rev.* **77**(8) 47–57.

Accepted by Hau L. Lee; received November 2002. This paper was with the authors 8 months for 2 revisions.