User Toolkits for Innovation

Eric von Hippel

MIT Sloan School of Management

Room E52-556, 50 Memorial Drive

Cambridge, MA 02141

(617-253-7155, evhippel@mit.edu)

Final version published in

Journal of Product Innovation Management, July, 2001

User Toolkits for Innovation

ABSTRACT

Manufacturers must accurately understand user needs in order to develop successful products – but the task is becoming steadily more difficult as user needs change more rapidly, and as firms increasingly seek to serve "markets of one." User toolkits for innovation allow manufacturers to actually *abandon* their attempts to understand user needs in detail in favor of transferring *need-related* aspects of product and service development to users along with an appropriate toolkit.

User toolkits for innovation are specific to given product or service type and to a specified production system. Within those general constraints, they give users real freedom to innovate, allowing them to develop their custom product 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. As the concept is evolving, toolkits guide the user to insure that the completed design can be produced on the intended production system without change.

Pioneering applications in areas ranging from the development of custom integrated circuits to the development of custom foods show that user toolkits for innovation can be much more effective than traditional, manufacturer-based development methods.

User toolkits for innovation

Introduction

New products and services must be accurately responsive to user needs if they are to succeed. 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. Deeper techniques, 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 such as custom products manufacturers are finding that conventional solutions are completely breaking down, 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 is being developed on the basis of patterns evolving in a few high tech fields. In this new approach, manufacturers actually *abandon* their increasingly frustrating efforts to understand users' needs accurately and in detail. Instead, they learn to outsource key *need-related* innovation tasks to their users, after equipping them with appropriate "user toolkits for innovation."

User toolkits for innovation first emerged in a primitive form in the 1980's in the high-tech field of custom integrated circuit (IC) design and manufacturing. In this field, the costs of not understanding user needs precisely and completely at the start of a product design project had grown to punishingly high levels by the 1980's, as custom IC products grew increasingly large and complex. Many errors due to 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.

In the early 1980's LSI Logic made a first, key move that led eventually to user toolkits for innovation: it released its own set of proprietary software design tools to its customers, so that the customers could design circuits for themselves. At that time, LSI was simply a small new venture, facing major entrenched competitors like Fujitsu in the field of custom integrated circuit design. However, as Wilf Corrigan, a founder of LSI, recalls: "When I talked to Yasufuku [a senior manager] at Fujitsu and told him that our plan was to put [custom IC design] software in the hands of the customers, he said, 'That is a brilliant strategy. If you do that and the software is good, you will win.'" [16].

Customer preference for using toolkits to carry out need-related design work for themselves proved to be so strong that, as Yasufuku had predicted, LSI quickly grew to be a major player in the custom IC market. Competitors were soon forced to follow LSI's lead and also introduce software that enabled users to "design their own." Results to date in the custom semiconductor field show development time cut by 2/3 or more for products of equivalent complexity and development costs cut significantly as well. Today, many billions of dollars of custom ICs designed by users and produced in the "silicon foundries" of custom IC manufacturers such as LSI are sold each year [3,5].

Although now only applied to the development of a few types of custom industrial products and services, we propose that user toolkits for innovation will eventually spread to most or all producers creating custom products or services in markets having heterogeneous customer needs. They will also provide the "design side" that is currently missing for users and producers of mass-customized products. In effect, user toolkits for innovation can provide users with true design freedom – as opposed to the mere opportunity to choose from lists of options that is currently offered by mass-customizers.[1]

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

Benefits from shifting design activities to users

At first glance, it does not seem to make much sense: Why should one be able to develop better products and services faster by transferring "need-related" work from manufacturer to user? After all, the same work is being done in both cases. However there are in fact great advantages having to do with (1) access to "sticky" user information and (2) with achieving faster, better and cheaper "learning by doing." The stickiness of a given unit of information is defined as the incremental expenditure required to transfer that unit from one place to another in a form usable by a given information seeker. When this cost is low, information stickiness is low; when it is high, stickiness is high. As has been shown elsewhere, users typically do have a great deal of need-related information about what they might want and about their situation that is sticky," and that therefore cannot be transferred to manufacturers at low cost. [14] Also, users typically do not know exactly what they want at the start of the design process. There is simply too much to know about the setting in which a novel product or service will be used for this to be possible. [13]

Of course, if a user does not know and cannot say precisely what he or she wants, a manufacturer of even custom products or services cannot expect to deliver the right solution the first time. Instead, an iterative process of design by trial-and-error typically ensues. First, the user gives the manufacturer need-related information in a specification for a desired custom product or service that is the best that he can do – but that is both incomplete and partially incorrect. The manufacturer then responds by supplying a custom solution that is only partially successful. The user then applies the product in the use setting, finds flaws, and requests corrections. This cycle continues until a satisfactory solution is reached [13,14].

During each cycle of the iterative process just described, the user is engaging in "learning by doing" [1,9]. Learning by doing in this case is the trial-and-error based process that begins when you design and build or buy a product or service that you *think* you want. When you then begin to use that product or service, you quickly learn that it is not quite right, and learn more about what you do really want. That is, you "learn by doing."

Why do problems with a product or service become crystal clear during early use, even though they were difficult or even impossible to anticipate prior to use? As was noted earlier, user needs and the user environment are very complex, and full of sticky, costly-to-transfer information. Details and subtle interactions cannot be fully captured in a specification – or even in the minds of user or manufacturer experts. Yet, these details do still exist – and any that cause problems *will* emerge when the new product or service is placed into use. As a simple example, consider the tale of the unfortunate boat builder who builds a boat in his basement, either forgetting the need to move the boat outside when it is finished, or assuming that his basement door is big enough to allow this. If the door from the basement is in fact too small, the setting *will* make the problem very clear the first time he actually tries to remove the boat.

In other words, novel products or services are specified and designed using models of a need and setting that are incomplete and partially inaccurate representations of the real world. But products or services must ultimately fit the real world, because that is where they will be applied. Adjustments are typically needed, and these are done by learning by doing, as problem-causing differences between the real world and the model arise and are resolved during use [13].

Learning by doing that draws upon sticky information about users' needs and situation cannot be avoided. In fact, it shouldn't be – after all, achieving a better fit between need and solution is a good thing to do! So the real issue facing the developer of custom products and services is how to make that process as efficient and effective as possible.

Shifting *need-related* development tasks to the user is the solution offered by user toolkits for innovation. This makes the process of developing new products and services better and faster for two reasons. First, the sticky, costly-to-transfer information about a user's need and detailed situation that must be drawn upon to accomplish those tasks is already located at the user site. Attempting to move that information to the manufacturer for design work by manufacturer-based designers is extremely difficult and costly. Using it where it is already located – at the user site – avoids this cost. Second, concentrating need-related design tasks *completely* within the user eliminates the need to shift problem-solving back-and-forth between user and manufacturer during the trial-and-error cycles involved in learning by doing.

To appreciate the major effect of these advantages, 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 be immediately 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 [4,10]. The advantages are similar in the case of product and service development. Thus, when custom integrated circuit design is carried out by entirely by manufacturers, users are only in a position to engage in learning by doing when a chip has already 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, as we shall see, 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.

Since user toolkits for innovation are intended for those who have sticky information related to product or service needs, the user for which a toolkit for user innovation is intended is not only or necessarily the "end" user of a product or service as traditionally defined. Firms can have *different* end users for different attributes of a product or service that they produce. For example, the end users of the *installation* features of an electrical light switch are electricians – and they are the ones with the best sticky information regarding how to install light switches effectively. On the other hand, traditional end users of switches are the ones with the sticky need information on what a light switch should look like and *do* once installed. Different toolkits can be designed for different types of end users of the same product or service when appropriate.

Toolkits – a way to transfer design capability to users

In principle, then, when "need-related" design tasks are assigned to users, times and costs can be compressed, and learning by doing based on sticky user information can be more seamlessly and effectively integrated into the design process. But the user is not a design specialist in the manufacturer's product or service field. So, how can one expect users to create sophisticated, producible custom designs efficiently and effectively? Manufacturers who pioneering in this field solve the problem by carrying out two major steps: (1) they "repartition" their traditional product or service development tasks in order to concentrate need-related problem-solving within just a few tasks – and then they assign those tasks to users; (2) they provide users with kits of design tools that can help them to carry out the design tasks assigned to them [12].

Executing the steps required for toolkit development is not necessarily either easy or cheap. Toolkit development can nonetheless pay manufacturers who repeatedly engage in custom product development, because it is a one-time cost. Once developed, a toolkit can be used by tens or hundreds or thousands of users to carry out unique custom product or service design projects. This is so because manufacturers of custom products tend to specialize in a given solution type, which they apply to the diverse application problems of many users. Therefore, the kit of tools and information users will need from any given manufacturer to solve their novel application problems will tend to be the same from problem to problem, involving such things as that manufacturer's process capabilities and constraints. In contrast, need information held by users will have novel components for each novel custom product or service desired. Therefore, if a manufacturer does not supply users with a toolkit and instead seeks to understand user needs, that manufacturer must invest anew to acquire novel information from users for *each* custom development project undertaken [12].

Repartitioning development tasks

In the conventional product and service development paradigm, problem-solving that draws heavily upon *need-related* information has typically been an element within many product and service development tasks. After all, if a manufacturer is to execute all the problem-solving in any case, it is irrelevant from the point of view of information-transfer costs whether many tasks or few require need-related information. However, if the goal is to transfer only need-related design tasks to users – and to make these tasks as few and simple as possible, then a manufacturer must typically rethink the way its new product and service development tasks are divided up.

This rethinking can involve fundamental changes to the underlying architecture of a product or service. Consider, for example, the repartitioning of tasks that was carried out by semiconductor manufacturers as they 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, since 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.

Rethinking of the custom design problem led to the insight that circuit elements could be made standard for all custom circuit designs, and that all customer need-related information about chip function could be concentrated entirely within the task of designing the unique configuration of the electrical "wiring" that lay on the top surface of the chip. Chips with an entirely new architecture, called gate arrays, were created to allow this repartitioning of tasks, and then the wiring design task *only* was outsourced to users along with a toolkit that would aid and guide them in its performance.

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 only – 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 fruits 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.

The repartitioning of product or service design tasks to obtain the end just described will sometimes require a major creative effort. Semiconductor gate arrays were not simply an obvious variant upon earlier custom semiconductor design practice, for example. As practice evolves and as examples multiply, generally fruitful approaches to the task may emerge. At this time, however, useful guidance currently available for those interested in task repartitioning is limited to a conceptual understanding of the goal being sought, plus a few illustrative examples.

Elements of a Toolkit

If a manufacturer outsources design tasks to users, it must also make sure that users have the information they need to carry out those tasks effectively, so that their added design costs are less than added benefits received. Manufacturers do this by investing in developing a toolkit for user innovation. Toolkits are not new as a general concept – every manufacturer equips its own engineers with a set of tools suitable for developing 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 modify furniture – in very novel or very standard ways. Others may have a kit of software tools needed to create or modify software. What is new, however, is integrated toolkits enabling users to create *and* test designs for custom products or services that can then be produced "as is" by manufacturers.

We propose that an effective toolkit 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

It is crucial that user toolkits for innovation enable users to go through complete trial-and-error cycles as they create their designs: Research into problem-solving has shown that trial-and-error is the way that problem-solving – including learning by doing – is done [e.g., 2,13]. For example, suppose that a user is designing a new custom telephone answering system for her firm, using a software-based computer-telephony integration (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, user toolkits for 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 [11]. 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, auto makers allow customers to select a range of options for their "custom" cars – 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 wide tire option did look great in the picture. But now that the car has been delivered, I discover that I don't like the effect on handling. Worse, I find that my car is too wide to fit into my garage!"

Similar disasters are often encountered by purchasers of custom computers. Many custom computer manufacturers offer a website that allows users to "design your own computer online." However, these websites do not allow users to engage in trial and error design. Instead, they simply allow users to select computer components such as processor chips and disk drives from lists of available options. Once these selections have been made the design transaction is complete and the computer is built and shipped. The user has no way to test the functional effects of his or her choices before purchase and first field use – followed by celebration or regret.

In contrast, a toolkit for user innovation approach would allow the user to conduct trial-and-error tests to evaluate the effects of initial choices made and to improve upon them. For example, a computer design site could add this capability by enabling users to actually test and evaluate the hardware configuration they specify on their *own* programs and computing tasks before buying. To do this the site might, for example, provide access to a remote computer able to simulate the operation of the computer that the user has specified, and provide performance diagnostics and related choices in terms meaningful to the user (e.g., "If you add x option at y cost, time to complete your task will drop by z seconds"). The user could then modify or confirm initial design choices according to design and preference and trade-off information only he or she knows.

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, user toolkits for innovation intended to create designs producible on automated machining centers can offer users access to that very large solution space.[2]

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" [7] production methods are restricted to combining "any frame from this list" of predesigned frames, with "any hinge from that list" of predesigned 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 in order 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

User toolkits for 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: "If I increase chip clock speed, I can reduce the size of my cache memory and…"

As 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 dying 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 on 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.

In general, steady improvements in computer hardware and software are enabling toolkit designers to provide information to users in increasingly friendly ways. In earlier days, information was often provided to users in the form of specification sheets or books. The user was then required to know when a particular bit of information was relevant to his or her development project, find the book and look it up. Today, a large range of potentially-needed information can be embedded in a computerized toolkit, which is programmed to offer the user items of information only if and as a development being worked upon makes them relevant [12].

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 pre-analyzed structural characteristics, that they can incorporate into their novel building designs. Similarly, designers of custom integrated circuits find it very useful to incorporate pre-designed 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 take-out 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: recipe development procedures based on food ingredients available to individuals and restaurants, and 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. And so, 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. But 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. So 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. Thus, a toolkit designed for Mexican sauce development would contain 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 interested in using the Nestle toolkit to prototype, for example, a novel Mexican sauce would receive a set of 20 to 30 precomponents, each packaged in a separate plastic pouch. They would also be given instructions for proper use. The chefs will find that each component differs slightly from the fresh components he or she is used to. But 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, field testing by Food Product Development Department researchers showed that adding the “error-free translation” feature to toolkit-based design by users can reduce 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.

Discussion

To this point we have explored why user toolkits for innovation can be valuable, and have developed the contents of a toolkit. We now conclude by discussing conditions under which toolkits will offer the most value; the type of user that will want to employ toolkits; how users and manufacturers both contribute to toolkit development; and the impact of toolkits on the competitive position of manufacturers.

Where user toolkits offer the most value

User toolkits for innovation are applicable to essentially all types of products and services where heterogeneity of user demand makes customization valuable to buyers. In this article we have illustrated this point via examples of toolkits being used to help users design custom integrated circuits, custom telephony services, custom hairstyles, and custom foods – quite a range. Note that toolkits can be applied to custom products produced in relatively large volumes, such as custom integrated circuits, or to products designed for single unit production, such as products produced by mass-customization production methods. In the latter case, as was mentioned in the article introduction, user toolkits for innovation can supply the "design side" that is missing from today's mass customization practices.

User toolkits for innovation can be applied to both physical goods and well as information goods and associated services such as custom telephony software and the services it generates. Also, toolkits can be used to design custom physical and information products or services that are then produced by a manufacturer – *or* are produced directly at the user's site.[3] Other things being equal, it is more effective from the point of view of complete avoidance of iterative problem-solving that bounces back and forth between user and manufacturer to produce actual prototype products or services at the user site. The user seldom knows all information about the use environment that will prove to be relevant to the design, and so cannot enter it into design simulations. As a result there is typically an element of trial-and-error design that is only carried out by learning by doing when the actual physical device or service is present in the actual – as opposed to simulated – user environment.

Within the arena of custom products and services, we propose that toolkits will deliver the greatest value when users have need information that is sticky, and when they must engage in learning by doing to clarify what they really want. This proposal is in line with studies of the impact of sticky information on the sources of innovation [6]. These conditions will hold especially strongly, we suggest, when the innovation at issue involves functional novelty as opposed to improvement along some wellunderstood dimension of merit such as cost or speed. For example, designing a new software product to do something novel can involve a lot of sticky information and trial-and-error learning by users before it is gotten right. This was the case with Lotus Notes when that groupware product was new. On the other hand, much less user-related sticky information or learning by doing was required when the user need was simply to make Lotus Notes work faster. From yet another perspective, we reason that user toolkits for innovation will tend to be most frequently applied and therefore most valuable when the rate of change in a market is high – meaning that information on user needs quickly grows obsolete and must constantly be reassessed.

User toolkits for innovation 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 product or software with the same skill as can a human designer. For example, a design for an ASICs 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. As a result, a full-custom design will be smaller and will run faster than will an equivalent design produced with the aid of a toolkit for user innovation.

Toolkits are not for all users

The design freedom provided by user toolkits for innovation may not be of interest to all or even to many users of a given type of product or service. Users must have a high enough need for something different 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.

Users who do end up using a toolkit will often be "lead users," whose present strong need foreshadows a general need in the marketplace. Manufacturers may well find it valuable to somehow acquire the generally useful improvements made by these lead users, and supply them to the general market in the "default" version of the product generally offered. This was the pattern followed by the Technicon Corporation with respect to its clinical chemistry autoanalyzer products, for example. Information on improvements made by clinician-users was actively collected by that company, and innovations of value to many users were incorporated into analyzers sold to the general market [15]. The pattern is also visible in the case of open source software products such as Apache server software. Here innovations developed by users are screened in some way, and the best are incorporated into the "official" version of the software, which is then generally distributed [8].

Toolkit design – a joint user/manufacturer effort

We have said that manufacturers that offer user toolkits for innovation to their customer 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 need 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 solution dimensions 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 distribute existing in-house design toolsets as a first-generation toolkit for user innovation as was done by LSI. 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 (demand!) the additional ingredients and capabilities they need to implement their novel designs. Manufacturers can then improve their toolkits by responding to these explicit requests for improvement. And/or they can wait until impatient lead users actually create and test and use the toolkit improvements they need. 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 are distributed to the general community of users.

Competitive advantages of toolkits for manufacturers

Toolkits can create competitive advantages for manufacturers 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 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 ASICs industry. The initial toolsets revealed to users by LSI and rival ASIC producers were producer-specific. Over time however, specialist tool design firms such as Cadance developed toolsets that could be used 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 be eventually 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 user toolkits for innovation will eventually be adopted by most or all producers creating custom products or services in markets with heterogeneous customer needs. As toolkits are more generally adopted, the organization of innovation-related tasks seen today especially in the field of custom integrated circuit development will spread, and users will increasingly be able to get *exactly* the products and services they want – by designing them for themselves.

References

1. Arrow, Kenneth J. "The Economic Implications of Learning by Doing." Review of Economic Studies 29: 155-73 (June 1962).

2. Barron, Jonathan Thinking and Deciding. New York: Cambridge University Press, 1988.

3. Chakravarty, Dev "Marketing ASICs" in Norman G. Einspruch and Hilbert, Jeffrey L., Editors Application Specific Integrated Circuit (ASIC) Technology. San Diego, California: Academic Press 1991, p 31, table 1.

4. Levy, Steven, "A Spreadsheet Way of Knowledge" Harpers 269, No 1614, pp 58-64 (November 1984).

5. McClean, William J., ed Status 1995, Scottsdale Arizona: Integrated Circuit Engineering Corporation 1995 pp.4-13.

6. Ogawa, Susumu "Does sticky information affect the locus of innovation? Evidence from the Japanese convenience-store industry." Research Policy 26 pp.777-790 (1998).

7. Pine, Joseph B. II. Mass Customization: The New Frontier in Business Competition, Cambridge, MA: Harvard Business School Press 1993.

8. Raymond, Eric S. The Cathedral and the Bazaar Sebastopol, CA: O'Reilly and Associates 1999.

9. Rosenberg, Nathan. Inside the Black Box: Technology and Economics. New York: Cambridge University Press, p. 131 1982.

10. Schrage, Michael. Serious Play: How The World's Best Companies Simulate to Innovate, Cambridge, MA: Harvard Business School Press p. 38, 2000.

11. Thomke, Stefan. "Managing Experimentation in the Design of New Products," Management Science, vol. 44, No. 6 pp. 743-762 (June 1998).

12. von Hippel, Eric "Economics of Product Development by Users: The Impact of "Sticky" Local Information" Management Science, vol. 44, No. 5 pp. 629-644 (May 1998).

13. von Hippel, Eric and Marcie Tyre "How "Learning by Doing" is Done: Problem Identification in Novel Process Equipment." Research Policy 24 pp. 1-12. (January 1995).

14. von Hippel, Eric. "Sticky Information" and the Locus of Problem Solving: Implications for Innovation" Management Science 40, no.4 pp. 429-439 (April 1994).

15. von Hippel, Eric. The Sources of Innovation New York: Oxford University Press Chap 7 1988.

16. Walker, Rob with Nancy Tersini. Silicon Destiny: The Story of Application-Specific Integrated Circuits and LSI Logic Corporation Milpitas, CA: C.M.C. Publications 1992, p.80

Endnotes

[1] "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. [7]. Today, producers supplying mass-customized products typically only allow customers to mix and match from predesigned lists of options. Thus they may offer users who want to design their own custom eyeglasses only the possibility of combining "any frame from this list" of predesigned frames, with "any hinge from that list" of predesigned hinges, and so on. Customers who want to stray beyond the proffered options are typically told, "Sorry, we can't supply that. Any new option needs to be carefully designed before it can be manufactured by mass-customized production methods." In other words, in today's practice the cost of *producing* unique items via mass customization has come down, but the cost of *designing* unique items -- those not assembled from preexisting design modules – has not. User toolkits for innovation can provide "the design side of mass-customization" by creating a way to offer user-designers significant design freedom while at the same time insuring that the designs they create can be produced on the intended production system.

[2] Note, however, that current computer-aided design and manufacturing software (CAD-CAM) is not equivalent to a user toolkit for 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.

[3] For example, the custom integrated circuit design practice we described earlier involved circuits that were designed and subjected to trial and error testing via computer simulation at the user site, with production then being carried out at a manufacturer's silicon foundry. An alternative customization process involves the creation of "field programmable logic devices." These are shipped to the user as actual integrated circuits that are packaged and externally complete – but with the internal electronics in an unfinished, uncustomized form. Special equipment then allows the user to transfer the custom design he or she has created using a user toolkit for innovation into the unfinished semiconductor to create a finished, directly usable custom integrated circuit. Similarly, toolkits for software or software-based services can enable users to create custom software in finished form directly at the user site – there is no need to send it back to the manufacturer to be "produced."