Toolkits for User Innovation:
The Design Side of Mass Customization

Eric von Hippel

Sloan Working Paper # 4058 February 1999

ABSTRACT

Today, purchasers of many types of mass-customized products have only a very limited freedom to customize. Typically, they are restricted to making selections from a list of options offered by a manufacturer. Toolkits for user innovation greatly expand users' ability to design new options and features – and so enable users to get more exactly what they want. Toolkits usually consist of "user-friendly" design software that enables users to carry out complete cycles of trial-and-error learning on their own, using their customary design language and skills. During the design process, the toolkit software is constantly checking that a users' evolving design stays within the bounds of the producible. Completed designs are automatically translated into the format needed by the intended production process.

Toolkits for user innovation have been implemented in a few advanced fields such as custom integrated circuitry with good customer acceptance. When compared to custom designs created by manufacturer-based designers, designs created by users with the aid of toolkits are likely to be developed more quickly, and are also likely to provide a better fit to user needs.

Toolkits for User Innovation: The Design Side of Mass Customization

Introduction

Standard products and services can't offer an excellent fit to the needs and wishes of most consumers – there is simply too much variation in consumer preferences. This means that many consumers are at least somewhat unhappy with what is offered in the marketplace. It also means that many users would be willing to pay more for a product or service that really does meet their unique needs precisely.

Despite the apparent opportunity for profit, volume producers traditionally had no realistic hope of serving individual users with unique needs. Volume production meant mass-production processes, and mass production processes meant uniform products and services. Relatively recently, however, the development of "mass-customization" has broken this iron rule. Mass customization involves computerized process equipment that can be adjusted instantly and at low cost. Such equipment can produce one-of-a-kind products at near mass-production costs. Today, producers using mass-customized production can invite users to specify one-of-a-kind products ranging from eyeglass frames, to automobiles to Barbie dolls. These custom designs can then be produced in single-unit quantities at costs that are reasonably competitive with the costs of similar, mass-produced items (Pine, 1993).

Mass customization has proven to be a very successful approach for many firms, and many mass-customized products have been very successful in the marketplace. But more needs to be done if mass customization is to reach its full potential. When one looks closely, the implicit promise that a user can get exactly the product that he or she wants from today's mass-customization methods fades significantly. The reason is that the design freedom available is typically very limited. For example, most mass customizers today restrict their customers to specifying unique combinations of options from a list. Thus, users who want to design their own custom eyeglasses are essentially restricted to combining "any frame from this list" of predesigned frames, with "any hinge from that list" of predesigned hinges, and so on. Similarly, car buyers who want to design a custom auto to serve their unique needs are only offered the opportunity to select from a list of manufacturer-designed options.

But what if you want a unique product that cannot be achieved by combining available options? For example, suppose you want to incorporate a tiny digital watch into your custom, "made just for you" eyeglass frames? Or, suppose you are avid about ice fishing and want to incorporate a porthole into the floor of your minivan in order to use it as a mobile ice fishing shelter? These are both perfectly reasonable ideas, and may solve strongly felt needs for the user making the request. Nonetheless, today's answer from the mass-customized producer is typically, "Sorry, we can't supply that. Any new option needs to be carefully designed before it can be manufactured by mass-customized production methods. We don't think that there will be enough demand for what you want to justify our investing in the design work required." In other words, 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.

To truly fulfill mass customization's implicit promise to give each user a unique, "just-right" product or service, mass customizers must learn how to change the economics of custom design – the "design side" of mass-customization. Interestingly, our research shows that firms in a few fields, such as custom software and custom integrated circuits, have learned how to do precisely this. Specifically, we have found that advanced firms in these fields have changed the economics of the design side of mass-customization methods by learning how to transfer a capability to design truly novel customized products and services to their users. They have done this by equipping their customers with application-specific, software-based "toolkits for user innovation." Where this has been done, the result is much faster design of custom products and services that offer a better fit to unique customer needs.

The benefits of this approach from the point of view of customer value can be so great that, when the approach is pioneered by one supplier in a field, competitors often find that they must follow suit. For example, in the field of custom integrated circuit design, LSI Logic was first to introduce toolkits for user innovation to the marketplace in the early 1980's. At the time, LSI was simply a small new venture, facing major entrenched competitors like Fujitsu. Wilf Corrigan, a founder of LSI, recalls that: "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." "Why don't you do that?" I asked. "Our software is so valuable that if we expose it to outsiders they will steal it," said Yasufuku (Walker 1992 p.80).

LSI Logic went ahead and provided design toolkits to its customers. Customer preference for this approach proved to be so strong that the firm quickly became a major player in the market. Competitors were soon forced to follow LSI's lead and also introduce toolkits for user innovation. Today, billions of dollars of user-designed custom ICs are sold each year by custom IC suppliers.

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 the work of custom design from supplier to user? After all, the same design work is being done in both cases. However there is in fact great advantage, and it has to do with achieving faster and better and cheaper "learning by using." (Rosenberg 1982)

Learning by using 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 using. Such learning is needed during product and service development because, typically, a user wanting a custom solution does not and cannot know exactly what he or she wants at the start of the design process. There is simply too much to know about a need and the setting in which a novel product or service will be used for this to be possible.

Of course, if a user does not know and cannot say precisely what he or she wants, a supplier of 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 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.

MFR ACTIVITY	USER-MFR BOUNDARY	USER ACTIVITY
	←	user draws on local need information to specify desired product or service
manufacturer draws on local capability information to develop prototype responsive to specifications	→	
	←	user evaluates prototype, drawing on local need information and changes specifications regarding what is desired
manufacturer iterates until user satisfied		user iterates until satisfied

Figure 1: Iterative Problem-Solving Pattern Often Encountered in Product and Service Development

Sidebar: <u>Anatomy of Learning by Using</u>

Why can't developers get it right the first time? And why do problems with a custom solution become crystal clear during early use, although they were difficult or even impossible to anticipate prior to use? The reason is that the user need and 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 supplier experts. Yet, these details do still exist – and any that cause problems <u>will</u> 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 <u>will</u> 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 using, as problem-causing differences between the real world and the model arise and are resolved during use.

The trial-and-error cycle involved in learning by using can be very time-consuming and costly. For example, von Hippel and Tyre (1995) studied the learning by using cycle that occurred after two novel custom production machines had been specified and purchased by a computer manufacturer from a custom machine supplier. Both machines had been built to the user's specification and both had been tested for error-free functioning before shipping. Nonetheless, during the first year of use, the customer was found to have requested 22 significant, non-routine repairs and improvements to machine functioning! The costs in terms of engineering time and lost production were quite significant.

Learning by using 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 custom design to the user makes that process better and faster for two reasons. First, there is a great deal of "sticky," costly-to-transfer information about a user's need and detailed situation that must be drawn upon to design a custom product or service. That information is generated at the site of the user. Moving it to the supplier for design work by supplier-based designers is extremely difficult and costly. Using it where it is already located – at the user site – avoids this cost. Second, concentrating the work of custom design completely within the user eliminates the need to shift problem-solving back-and-forth between user and supplier during the trial-and-error cycles involved in learning by using (von Hippel 1994).

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 by 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. The advantages are similar in the case of product and service development, although the savings in time and cost may be even greater. For example, suppose that a manufacturer designs a reasonably complex "full-custom" integrated circuit chip for a user. The user will be sent a prototype chip to try, will try it, and then will very probably detect problems and request changes – an outcome that we would expect on the basis of learning by using, and the presence of sticky information known to the user, but not the manufacturer. In this case, a second prototype that incorporates the requested changes may require two months to develop and cost $100,000 in design charges.

These long cycle times and high costs are in part due to the need to transfer sticky information from user to supplier, and in part due to the coordination needed between user and a supplier to execute them. For example, a revision requested by a user might have to wait until the supplier can assign the designer who executed the original design – now busy on another project – to carry out the requested rework. On the other hand, if the design work is carried out entirely by the user, many of these costs are avoided. Learning by using via trial-and-error still occurs, of course, but the cycle time is much faster because the complete cycle is carried out at a single site. (Indeed, since cycle times and costs are sharply reduced, users may elect to increase the number of trial-and-error cycles carried out in order to get a better result, and still finish the project faster than an equivalent project involving user-manufacturer coordination during design [Thomke 1998].)

Toolkits – a way to transfer design capability to users

In principle, then, when the user does the complete design job, times and costs can be compressed, and learning by using can be more seamlessly and effectively integrated into the design process. But the user is not a design specialist in the supplier's product or service field. So, how can one expect users to create sophisticated, producible custom designs efficiently and effectively?

Suppliers who have pioneered in this field have solved the problem by developing sophisticated, software-based kits of design tools explicitly for customer use. These toolkits are designed to achieve four important objectives. First, they enable the user to carry out complete cycles of trial-and-error learning, drawing on sticky, local information not known to the supplier. Second, users can operate them by employing their customary design language and skills – making them "user friendly" in the sense that users do not need to engage in much additional training to use them competently. Third, they 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. Fourth and finally, the toolkit software is designed to ensure that custom products and services designed by users will be producible on supplier production equipment without requiring revisions by manufacturer-based engineers (von Hippel 1998).

Toolkits should have all four of these characteristics to truly shift substantial design freedom and capability from the supplier to the user. The first two characteristics, however, are the most interesting to explore in order to fully understand the wide potential of toolkits for user innovation.

It is crucial that toolkits for user 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 using – is done. For example, suppose that a user is designing a new custom telephone answering system for her firm, using a software-based 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.

When mass customizers do not supply their customers with toolkits that enable them to draw on their local, sticky information and engage in trial-and-error learning during custom product design, a customer must actually order the product and have it built to learn about design errors – typically a very costly and unsatisfactory way to proceed. For example, although auto makers allow customers to select a range of options for their "custom" cars, 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 furniture. Custom furniture manufacturers often tell purchasers that "We can make anything – just tell us what you want." However, they do not provide any tools, beyond salesroom samples and fabric swatches, that can allow the customer to learn precisely what he or she really wants via trial-and-error before ordering. If the customer is lucky, the first trial design will satisfactory. If not, very unwelcome lessons will be learned too late: This couch turns out to be way too big for the room! This fabric (which looked so good on the swatch) doesn't fit with the wallpaper!" etc.

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 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 incorporate 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, they 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 translates the user's logical design into a different form, the design inputs required by the IC manufacturer's mass-customization production system. In this translation, the circuit designed by the user is converted into a design made up of transistors and other electrical components to be constructed upon the surface of a silicon wafer during the manufacturing process. Because of this "user-friendliness," the user need know nothing about the manufacturer's design language or production process in order to design a successful chip – the toolkit takes care of these matters for him.

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 supplier-based experts.)

Consider, for example, 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 may be very 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 user 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. As the user works, toolkit software warns the user if her custom design is going outside of the bounds that a professional hairstylist can actually create. Then, when the user is satisfied with the design - and just as in the case of integrated circuit design - the user's completed design can be translated into technical hairstyling instructions in the language of a hairstyling specialist – the intended "production system" in this instance.

Implementing toolkits for user innovation

Providing a toolkit for user innovation to your customers can yield great profits – but it is by no means a trivial undertaking. To implement the approach, suppliers need to understand what their non-specialist user-customers already know, and then create a design process and toolkit that largely fills in what they do not know in order to create a complete and producible outcome. To this point, the toolkit approach has largely been developed in fields where design is already customarily done by suppliers using software-based design systems. Examples of such fields are integrated circuit design, design of computerized telephony systems, etc.. In such cases, the task of switching to a toolkit for user innovation approach typically begins by creating a prototype toolkit that is a "user-friendly" version of the design tools already in use. In fields such as hairstyling, software design systems are not customarily used. Toolkit design efforts are nonetheless going on, but more development effort is required in such cases, because the needed software design systems must be designed from scratch.

In either case, a supplier engaged in toolkit development proceeds by first releasing a prototype toolkit to a few "lead users" that have a high need for user-based design. The learning by using experienced by these lead users then can provide key information needed for an improved version that can be more generally distributed. Indeed, lead users will often actually create and test the needed improvements for the supplier rather than simply suggest them!

Toolkits for user innovation will typically give most advantage to users and profit to suppliers when users need products or services for novel applications, and when they need them a lot.

Novel applications benefit from toolkits for user innovation because, typically, a lot more sticky user information is needed to design a product or service for a novel application than for a familiar one. For example, machines used in logging, such as the log skidders that pull cut trees out of the woods, often were based on machines originally designed to be used in construction or some other field. When machine builders were first approached about applying similar machines to logging, the builders did not have a clue as to how the machines should be redesigned to serve successfully in the woods. Instead, the complex "sticky" information that would be required to design machines for this new application resided in the minds and the use situation of logging company personnel. In contrast, machine builders had a much better idea as to how to build the next, more powerful version of their existing construction machines, because much of their painfully acquired trial-and-error learning applied to this situation. That is, they did not have to go out and learn much that was new in order to satisfy existing customers who were working their machines in existing, known applications.

Users that get significant economic benefit from getting custom products or services fast are the most promising toolkit customers. For example, new electronic products often incorporate custom ICs, and cycle times for new electronic products are short. So custom IC users simply can't wait for the traditional, iterative trial-and-error design refinement process carried out between user and supplier. They therefore are good customers for a design approach based upon toolkits for user innovation.

Suppliers potentially interested in trying a toolkit-based approach to custom product or service design should be aware of an important business model issue that differs from "business as usual." This issue arises because suppliers that both design and produce custom products and services capture profit from both their design capabilities and their production capabilities. A switch to user-based design for customization can affect their ability to do this in the long run. In the short run, however, profits will probably increase. When a supplier begins to provide toolsets for user innovation to its customers, the customers take on some of the design work that the supplier formerly had to do in-house. The customers will also be willing to pay for the privilege, because of the better, faster custom designs that they get from the toolkit approach. The first toolkits introduced into a field are also likely to be supplier-specific – so that your customers must come to you to produce what they have designed using your toolkit.

In the longer run, however, independent tool developers can emerge that will create toolkits for user innovation capable of designing products that can be produced by any one of many competing suppliers. (In fact, this is precisely what happened in the custom IC industry. The initial toolsets released to users by LSI and rival 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 any of a number of vendors.) When this happens, the tie that traditionally existed between design and production ("I only design what I build") is broken, and firms must learn to profit from toolkits and/or production as independent capabilities.

If any firm introduces toolkits to a field favorable to their use, customers will move to it and competitors will be driven to follow. Therefore, a firm's only real choice with respect to adopting toolkits for user innovation when conditions favor their use is the choice between leading or following. It seems to us that leading is better!

References

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

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

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

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

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

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

Rob Walker with Tersini, Nancy (1992) Silicon Destiny: The Story of Application-Specific Integrated Circuits and LSI Logic Corporation C.M.C. Publications, Milpitas, California

Date Due

APR		
	APR 1	