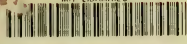


MIT LIBRARIES



3 9080 00567356 8





**WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT**

**Shifting Product and Service Prototyping
To Users:
An Innovation Process Advantage?**

Eric von Hippel

June, 1989

WP# 3032-89-BPS

**MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139**

**Shifting Product and Service Prototyping
To Users:
An Innovation Process Advantage?**

Eric von Hippel

June, 1989

WP# 3032-89-BPS

I wish to thank Mr. Bradley Feld, Doctoral Candidate at the MIT Sloan School of Management, for his important contributions to the ideas contained in this paper.

MAY 10 1989

Shifting Product and Service Prototyping To Users: An Innovation Process Advantage?

1.0: Introduction

The traditional division of innovation process activities between users and manufacturers assigns to users the role of "having needs". The role of the manufacturer is then to develop the new products and services responsive to those needs. In this paper, I begin by arguing that this traditional division of labor is inefficient, based on what we know about "task partitioning". I then offer an alternative, more efficient partitioning in which the user takes over more of the new product and service development task than has been traditional. I next explore trends in software development that make such a transfer increasingly feasible in many fields. Finally, I note some implications of the possible shift of new product and service development to users for the marketing research activities of firms.

2.0: Partitioning the Innovation Process Between Users and Manufacturers

An innovation project of any magnitude is divided up into a number of tasks and subtasks (a process I term "task partitioning") that may then be distributed among a number of individuals, and perhaps among a number of firms. When the partitioning process is complete, the component tasks and their interfaces are specified - implicitly or explicitly - so that all will fit and work together to form the total project when combined. Such specifications say in effect: "This is the nature of Task X. These are the inputs it will or can receive from other parts of the project; and these are the outputs it must provide to specified points in the project at specified times."

Innovation managers have a choice as to how they partition the tasks involved in an innovation project. The division of labor between the user and the

manufacturer in the development of new products and services represents such a choice - although one so traditionally and automatically made that its optional character may not be immediately apparent. This traditional partitioning assigns to users the role of "having needs". The role of the manufacturer is then to develop the new products and services responsive to those needs. Is this partitioning efficient? I propose that it is not, and will develop this point by first considering general criteria for efficiency in task partitioning, and then returning to assess this specific partitioning decision.

2.1: Problem-Solving Independence:

A Criterion for Task Partitioning

In 1964, Christopher Alexander, an architect, proposed that the overall designs of houses or communities could be improved if they were made up of subsystems that could be adjusted relatively independently. Traditional designs had this characteristic, he said. He then argued that modern designers must strive to specify subsystems in their projects so that they were independent in this sense, lest the design problems they face become so complex as to be insoluble. (1)

In 1973 Herbert Simon made a similar argument with respect to "decision-making" tasks as follows:

"The division of labor is quite as important in organizing decision-making as in organizing production, but what is being divided is different in the two cases. From the information-processing point of view, division of labor means factoring the total system of decisions that need to be made into relatively independent subsystems, each one of which can be designed with only minimal concern for its interactions with the others. The division is necessary because the processors that are available to organizations, whether humans or computers, are very limited in their processing capacity in comparison with the magnitude of the decision problems that organizations

face. The number of alternatives that can be considered, the intricacy of the chains of consequences that can be traced -- all these are severely restricted by the limited capacities of the available processors." (2)

The desirable criterion put forward by both of these authors can be summarized as: Set the boundaries between subsystems so as to minimize problem-solving that must be carried out across such boundaries. I develop this criterion with respect to innovation project task partitioning elsewhere (3). For present purposes, the reader may find it useful if I simply convey an intuitive feeling for the importance of specifying innovation tasks so as to reduce their problem-solving interdependence, by means of two very simple and schematic examples. Each specifies an innovation project and then suggests two alternative ways to divide the project into two component tasks. These alternatives differ with respect to problem-solving interdependence between tasks and - as a consequence I suggest - also appear to differ with respect to the efficiency with which they can be carried out.

First, consider how one might partition the project of designing an airplane. In fact, of course, such a project would be partitioned into thousands of tasks. But, for present purposes let us assume that it will be partitioned into only two tasks, each to be undertaken by a different firm. The two alternative partitionings I propose we compare:

- "Firm X is responsible for the design of the aircraft body and Firm Y is responsible for the design of the engine,"

and:

- "Firm X is responsible for designing the front half of the aircraft body and engine, and Firm Y is responsible for the back half of each."

Taken together, each of these proposed partitionings has the same project

outcome - a complete aircraft design. But the two differ greatly with respect to the interdependence of the two tasks specified. The second alternative would require a much higher level of problem-solving between the two tasks. For example, many design decisions affecting the shape of the "front half" of an aircraft body could not be made without forcing related changes on the designers of the back of the body and vice versa: The two halves cannot be considered independently with respect to aerodynamics. In contrast, the detailed design of a complete aircraft engine is much less dependent on the detailed design of a complete aircraft body. As a direct consequence, I suggest, engineers would think the former partitioning far more efficient than the latter. Indeed, faced with the latter proposed division, experts would be likely to throw up their hands and say, "It can't be done that way".

As a second example, consider how one might partition the project of designing the interiors of two rooms between two interior decorators. One might assign each room to each decorator; one might assign one-half of each room to each. Again, the same work is to be accomplished in each instance. Only the way it is divided up has been changed.

In this example, the idea of two interior decorators each designing one-half of a room probably seems absurdly inefficient to the reader. And again, I propose that this is because of the need for between-task problem-solving that is inferred. That is, it seems reasonable that a solution devised by one decorator and implemented on one side of a room must cause the second artist to make responsive adaptations on the other side of the room if a satisfactory total design is to result.

We can see that it is the need for problem-solving across tasks that makes these partitionings seem inefficient by slightly changing the nature of the task in this second example. Suppose that problem-solving is clearly not involved in the room-design project. For example, suppose that the physical task is the same - two interior decorators are each assigned one-half of a room to design - but suppose that the decorators work for a hotel chain and proceed according to a strict formula. In

that case, asking each decorator to design half a room might be a perfectly acceptable, and possibly even efficient, partitioning of the task. For example, one decorator could specialize in applying the formula to window decorations, and one could specialize in applying it to room furnishings.

2.2: Assessing User and Manufacturer Innovation Process Roles in Terms of Problem-Solving Independence

Recall now that the traditional partitioning of innovation process tasks between users and manufacturers assigns to users the task of "having needs", and to manufacturers the task of assessing these needs and developing responsive products. I propose that this partitioning is a source of significant problems for innovation projects employing it, because these tasks are typically (but not always) highly interdependent in terms of problem-solving. The high interdependence comes about because, as is suggested in Figure 1, the ability of manufacturers to accurately understand user need without iteration is poor.

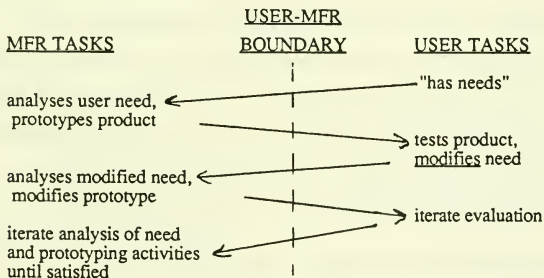


Figure 1: High Task Interdependence Characterizes Traditional Partitioning of New Product and Service Development Between User and Manufacturer

Why do I suggest that it is so hard to accurately understand many user needs without iteration and associated learning? Because individual industrial and consumer products or services are only components in larger usage patterns that may involve many such. If a proposed new product or service is simply a "plug-compatible" substitute for an existing component in such a pattern, it may be easy for a manufacturer to accurately perceive user need without the need for cycles of try and repeat. But, consider the difficult problem-solving steps a manufacturer - or a user - must go through to "accurately understand the user need" for a new product, process or service that is not a plug-compatible substitute for one that already exists. First, one must identify the existing multiproduct usage patterns in which the new product might play a role. Then one must evaluate the new product's potential contribution to these. (For example, a change in the operating characteristics of a computer may allow a user to solve new problem types if he makes related changes in software and perhaps in other, related products and practices. Similarly, a consumer's switch to microwave cooking may well induce related changes in food recipes, kitchen practices, and kitchen utensils.)

Next, one must invent or select the - possibly novel - usage patterns that the proposed new product or feature makes possible, or makes more desirable, and evaluate its likely utility in these. Finally, since substitutes exist for many multiproduct usage patterns (e.g., many forms of problem analysis are available in addition to the novel ones made possible by a new computer) one must estimate how the new possibilities presented by the proposed new product will compete (or fail to compete) with existing options.

Such a problem-solving task is clearly a very difficult one for either a user or a manufacturer analyst. It involves either very good thought experiments or, more realistically, developing a good combination of need and solution by a learning-by-doing cycle of experiment, modify and repeat.

If we now think about the traditional partitioning of the innovation task, we see that the tasks traditionally assigned to the user and the manufacturer are in fact highly interdependent with respect to problem-solving. The manufacturer must sense the user's best initial perception of his need, develop a responsive product and deliver it to the user for trial. The user must then experiment with the product and develop a different, more accurate understanding of his need. The manufacturer must then sense this new perception of need by the user and repeat the cycle until a suitable pairing of need statement and responsive product is identified.

3.0: Improving User-Manufacturer Partitioning of Innovation Tasks by Shifting Product and Process Development to Users

Given the above, how can we improve on the traditional partitioning of innovation tasks between user and manufacturer? One possibility is to shift the product development (at a minimum, product prototyping) capability to the user. Recall that this solution is based upon the following reasoning: Learning by doing means experimenting with solutions. Such experiment must be done by the manufacturer using users as subjects or by the user directly. In either case, the user must be involved because, after all, it is the fit to his systems we are attempting to determine and improve. And, as has been discussed earlier, if there is high problem-solving interaction between need specification and solution-trial or learning by doing, then one wants to get the activity into one locus - the user locus in this instance. Therefore, let us now consider: what is the realistic likelihood of shifting the locus of (prototype) innovation to the user?

It has been shown that, given sufficient incentive, users now develop products to suit their particular needs.(4) Thus, users of software often write their own application programs; scientists often design and build instruments to serve their special needs; firms often build the specialized process equipment needed in

their factories; creative cooks may develop their own recipes.

It has also been shown that user innovation can be affected by the provision of easier product development possibilities than that of designing a product from scratch. Thus, a firm which provided its medical laboratory customers with a modular clinical chemistry analyzer and thereby made user modification easier (less expensive) experienced more user modification activity that resulted in commercially valuable products than did a manufacturer of a competing, less modifiable instrument of equivalent function.(5)

Table 1: Developers of Best-Selling Tests For Two Competing Brands of Clinical Chemistry Analyzer

Technicon Analyzer	<u>% User</u> 74%	<u>User</u> 14	<u>Mfr</u> 5	<u>NA</u> 1	<u>Total</u> 20
Du Pont Analyzer	<u>% User</u> 0%	<u>User</u> 0	<u>Mfr</u> 18	<u>NA</u> 0	<u>Total</u> 18

Clinical chemistry autoanalyzer users and manufacturers explain the data summarized in Table 1 by noting that Technicon analyzers were much easier to adapt to new test development than were Du Pont analyzers. The cause of this difference lay in the design of each product's reagent handling system. Let me briefly describe that aspect of each brand's design to make the matter clear.

Technicon automated clinical chemistry analyzer models are based on a principle called "continuous flow analysis," and function much like miniature, continuous-process chemical plants. They consist of functional modules - e.g., pump modules, dialyzer modules, etc. - that are connected by plastic tubing.

Reagent is placed in bulk reservoirs and metered into the system as needed.

The Du Pont "aca" clinical chemistry analyzer uses reagents supplied in single-use, disposable, factory-sealed "test packs". These are quite complex. Each contains a plastic pouch divided internally into several sealed compartments which contain reagent quantities needed for a single execution of a particular test. The pouch itself is sealed to a plastic "header" that contains a serum inlet valve and, for tests that require it, a built-in chromatographic column. All chemical reactions required for a test occur inside the disposable test pack - the pack itself is never opened during its transit through the analyzer equipment.

On the basis of the above capsule descriptions, the reader may find it reasonable that users could experiment with novel test methods and equipment configurations by using Technicon equipment at a lower cost than could be done by using Du Pont equipment. Technicon modules may be purchased and connected up in a novel configuration. In Technicon equipment, desired novel reagents can be mixed up in bulk, placed in the machine's reservoirs, and the machine will meter out the proper amount of reagent(s) and serum needed for each test.

Setting up the same novel method on Du Pont equipment, on the other hand, would require buying empty test packs from Du Pont (empty packs without chromatographic columns are for sale: They have a standard use in machine calibration). The experimenter would then inject precisely measured amounts of reagent into selected compartments of each pack and reseal each compartment. If 1,000 tests were required for an experiment, he would have to perform these operations on 1,000 packs. This would clearly be a great effort - and the end result would be the accomplishment of a reagent proportioning task which Technicon equipment does automatically.

Bradley Feld has recently shown (6) that a radical reduction of the cost to users of development or prototyping products is in the offing due to recent developments in software. These advances directly lessen the cost of user

development of application programs. Indirectly, they lessen the cost of developing products or services that involve software in their development or use - a very considerable universe. Thus, the development and fabrication of hardware-based products is increasingly affected by means of computer-aided design and manufacturing software, and by software-operated process equipment ranging from numerically controlled machine tools to software-driven semiconductor fabrication equipment. And, the "hardware-based" products provided to end users increasingly contain a combination of software and hardware components. Examples range from the computer itself, to complex medical equipment such as the CAT scanner, to consumer products such as video games. And, of course, many services ranging from on-line reservation services to automated teller services are created and implemented largely through the development of new software application programs.

The advances in software behind this reduction in costs come from two sources: (1) the provision of programs to users containing multiple "optional" functions; (2) advances in "user-friendly" programming tools such as object-oriented programming. I will briefly discuss each in turn.

One way to lower the cost of user product development is to supply products that have lots of available options. The "tailorable through options" strategy has become vastly more practical recently in the field of software relative to the situation we find in hardware-based products.

In the instance of traditional, hardware-based products, it is emphatically not costless to provide users with options - that is, optional product "features". First, it costs a manufacturer an additional one-time cost to design each additional feature offered and, second, it costs an additional per-copy cost to install each additional feature in a product. The "all features provided" hardware product can also have an additional cost from the user point of view: Unwanted features can interfere with wanted features simply by their physical presence. For example, the presence of each not-wanted feature on a car, such as a convertible top or extra knobs on a radio,

could reduce the value of wanted features to the user by taking up space and reducing the convenience of the wanted features.

An alternative strategy of offering users a tailorable product by offering different versions of a product, each incorporating a different subset of features is possible, but it forces the user to justify his investment in a feature by knowing in advance that he will find the feature useful. This requirement works against the goal of offering the user tools for developing new products and services in a learning-by-doing mode ad lib.

In the instance of software-based products such as word processors, the possibility of costlessly offering users many optional features is much closer to reality. There are two reasons for this. First, while it still costs money to design each additional software feature, it does not cost additional money per copy to produce each additional feature. Second, menus and macros allow users to tailor a product offering all features, so that only the features a particular user wants need be visible. For example, the word processing program I use on my PC may have (indeed, does have) a mail merge feature. But, if it is properly designed, it will be totally invisible to me unless I invoke it, and therefore will not interfere with the functioning of the features I do want to use.

Users' ability to tailor software via a list of optional features clearly aids users in tailoring products to their particular needs in a learning-by-doing fashion. But what if the modification a user contemplates cannot be constructed out of the available options? In that case, programming advances such as object-oriented programming and fourth generation languages make the devising of new modifications "from scratch" easy. Thus, object-oriented programming provides the user with graphic images that represent common software functions. By simply linking objects in a way that graphically represents desired function, the user can construct a program to meet his needs.

4.0: Discussion: Some Implications for Marketing Research

If user innovation is indeed an increasingly common phenomenon, then manufacturer marketing research may be increasingly interested in studies of "lead users" that have been described elsewhere(7). In addition, marketing research may wish to encourage user innovation by making the product easier to modify in general(5), and/or providing more options of possible utility to users interested in modifying or adapting their systems. Either path can encourage the learning by doing that improves user (and manufacturer) understanding of user needs.

To illustrate the latter point, let me draw on a case reported on by Wendy Mackay (8). Mackay studied the behavior of a user of electronic mail before and after that user tried out certain mail screening features contained a a program called LENS. Prior to LENS, the user attempted to screen all of her mail for importance by visual scanning. The high time-cost of this solution left her always behind and concerned about missing something important. Eventually and reluctantly, the user tried a screening rule offered by LENS: Identify and segregate all mail addressed to the user by name (vs. mail addressed to "distribution list X"). Via the trial, the user found that (most?) mail she considered important fell into this category. Thus, she found this screening rule helpful - it considerably lessened the time she needed to devote to visual screening of mail - and so she adopted it.

Why did the easy availability of the screening function allow the user to learn more about her need in this instance? After all, a personal salutation screening rule could have been applied by the user in scanning through her mail by hand prior to its automated availability on LENS. However, in manual mode, the user had to actually look at the mail to implement this or other screening rules. Once actually looking at the mail, it would be low cost to apply some other rules as well (e.g., glance at content). Therefore, the user had no immediately obvious reason to test the effect of the personal salutation rule alone when screening mail manually since manual

application of the rule in isolation was unlikely to be cost-effective. As a result, she probably never would have learned about her need for such a rule absent its low-cost availability as an option offered by the software maker.

Changes in the traditional user-manufacturer interface with respect to new product and service development are clearly non-trivial matters. If the particular change that I propose in this paper is indeed feasible and offers the promise of increased innovation process efficiency, then it would seem an interesting matter to study and develop further.

References

- (1) Christopher Alexander, Notes on the Synthesis of Form (Cambridge, Mass.: Harvard University Press, 1964).
- (2) Herbert A. Simon, "Applying Information Technology to Organization Design," Public Administration Review 33: 268-78 (1973). (Reprinted with minor revisions as Chapter 14 in Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization, 3rd ed. [New York: Free Press, 1976]), 293.
- (3) Eric von Hippel, "Task Partitioning: An Innovation Process Variable," MIT Sloan School of Management Working Paper No. 2030-88, Cambridge, Mass., June 1988. Forthcoming in Research Policy.
- (4) Eric von Hippel, The Sources of Innovation (New York: Oxford University Press, 1988).
- (5) *Ibid.*, chap. 7, "Shifting the Functional Source of Innovation."
- (6) Bradley Feld, "The Growing Role of the User in the Development of Software," MIT Sloan School of Management Working Paper, forthcoming.
- (7) Glen L. Urban and Eric von Hippel, "Lead User Analyses for the Development of New Industrial Products," Management Science 34: 5 (May 1988): 569-82.
- (8) Wendy E. Mackay, "Diversity in the Use of Electronic Mail: A Preliminary Inquiry," ACM Transactions on Office Information Systems, October 1988.

Date Due

NOV 13 1989

APR 18 1990

APR 01 1991

MAY 10 1992

OCT. 27 1992

MAY 30 1994

MIT LIBRARIES DUPL 1



3 9080 00567356 8

