

Let Prototyping Drive Your Product Development Process

by Preston G. Smith CMC

Many firms base their product development efforts on phased development processes. The thrust of these processes is to stipulate the product requirements at the outset, then execute the process to provide those requirements – a 'say-what-you'll-do' then 'do-what-you-say' approach.

A difficulty facing product developers is that often neither we nor the customers are clear on the features desired for a given product at the outset of a project. Although it seems reasonable, the notion of frozen specifications when initiating development is often simply not manageable. Don Reinertsen, author of *Managing the Design Factory* and coauthor of *Developing Products in Half the Time*, has found through research that fewer than five percent of developers start designing with a complete set of specifications. Even this fortunate few find that the customer changes its mind during development, leading to the phenomenon of scope creep. Consequently, we need a process that allows us to learn how acceptable the product concept is as we go.

Prototypes are the perfect tool for learning while en route. What is a prototype? It is helpful here to think broadly and view prototype interchangeably with model, test, experiment, mock-up, or simulation. This new prototype-driven approach to product development was illuminated by Michael Schrage in *Serious Play* (Harvard Business School Press, 2000) and based on principles that you can follow by Stefan Thomke in *Experimentation Matters* (Harvard Business School Press, 2003). Let's explore some of their principles.

Although prototypes have been a part of product development for ages, only recently has their power multiplied to understand, communicate, and solidify action behind product options, largely due to computerized technologies that make prototypes far more affordable and accessible. Although the power now exists to use prototypes in new ways, new technologies are most often being used in the same old ways, leaving their true power to improve product development untapped.

Traditionally, you could only afford a few prototypes because they were expensive and slow to make. As a result, they appeared late in the process and were used mainly to ratify design decisions that had already been made. Being late and expensive encouraged the prototype builder to refine it as much as possible, which made it even later and more expensive.

Today, countless, simple but progressively refined prototypes are used very early in the process to help make design decisions. These could be to

- Decide among functional options (technical prototypes)
- See if a feature is intuitively clear (customer prototypes)
- Understand who might buy it (market prototypes)
- Explore appropriate business models (financial prototypes)
- Solicit supplier advice (sourcing prototypes)
- Elucidate productibility problems (manufacturing prototypes)

The key is to make plenty of prototypes, keep them each as simple as possible, and make

and assess them quickly. You do this by aiming each prototype at a specific question to be answered – what I call hypothesis testing. Keep each prototype as crude as you possibly can to resolve only its hypothesis. When a prototype has answered its question, toss it and plan the next round of prototyping.

The principle is alluringly simple, but executing it is not so easy – and this is why its adoption has been slow to date. I'll suggest some items to watch as you consider moving toward this mode of development.

- First, recognize that, if you are doing it well, the majority of your prototypes will be failures, illuminating a route down which you do not wish to proceed. This learning is valuable, but most of us do not want to expose our ignorance so visibly. Consequently, the organization must learn to cherish failure.
- If this did not deter you, keeping your prototypes simple – even crude – will be a challenge. We all like to burnish our work, and executives may believe that a sloppy prototype reflects sloppy thinking. For instance, the revered product development firm, IDEO is known for its prototyping effectiveness. But if you examine Tom Kelley's book about IDEO, *The Art of Innovation* (Doubleday, 2001), you will find only burnished prototypes among its many beautiful illustrations. Even IDEO has difficulty revealing an ugly prototype.
- Clearly, in order to base your flow of decisions on prototypes, you will probably have to re-engineer your development process so that prototypes can be built and assessed quickly and truly influence development decisions in real time. This means having a prototyping lab with plenty of capacity and no red tape in accessing it. It also means having a truly cross-functional team that can reach a decision on a prototype quickly and move on to the next round.

- While contemporary prototyping technologies are much faster and cheaper than previous ones, they can also have serious shortcomings. Frequently, they do not produce models with the fidelity of traditional methods. Consequently, you will need to integrate the new technology with the traditional one to preserve the strengths of each, and not simply replace the new with the old. This can be difficult, because the new and old technologies often have differing corporate cultures behind them.

- When building numerous prototypes, you will face prototyping strategy decisions. For example, should you build several prototypes in parallel during one round or string them out sequentially over several rounds? One method is faster, but the other is cheaper. The relative value of time versus prototype cost and the extent to which learning from one round influences the next will help you choose the proper strategy.

- Finally, when you are churning out prototypes, you will have to watch your overall prototyping budget. Usually the new technologies are so much cheaper that you can expect your total prototyping budget to decrease. But the corporate cost cutters may be alarmed when they see a tall pile of discarded prototypes. And some developers may continue to build the old, expensive prototypes, just to be sure.

In summary, if you live in a relatively static world in which you can define your product at the outset and then develop it according to plan without the plan changing, you should stick with phased development. Otherwise, you might consider a learn-as-you-go prototyping approach for greater agility.

Contact the Author: Preston Smith, Founder, New Product Dynamics at preston@NewProductDynamics.com

Reprinted with permission from *Sopheon News*, Tuesday, October 7, 2003, page 3.