Leverage the Cycle-Time Capability of Your Rapid Prototypes

Preston G. Smith

Rapid prototyping is a technology that undoubtedly reduces the time it takes to get a product to market. However, this article argues that rapid prototyping, in isolation, may not save as much time within the whole product development cycle as we would like to think. As a result, the author looks at how typical cycle-time problems in a complete development process can be overcome and identifies where huge potential time savings can be made, specifically looking at the fuzzy front end of the process, and the transition to manufacturing.

Reprinted with permission from *Time-Compression Technologies* (North American Edition), December 1998 (vol 3, no 7). Also published in *Time-Compression Technologies* (European Edition), March/April 1999 (vol 7, no 2), pp 50–57.

Driven by extraordinary advancements in computer performance over the past decade, rapid prototypes' often claim a speed advantage of 2X to l0X over previous methods. Measured in isolation, these claims are completely correct. However, in the context of the overall development cycle—measured from the time an opportunity for a new product arises until the company is shipping this new item—the effect of rapid prototyping (RP) is often negligible. Unfortunately, management and the owners of the business, who make the ultimate decisions on RP technologies, are far more interested in getting a new product to market quickly than they are in the burst speed of a particular technology.

This paper explains how to convert time-compression technologies into real business advantages by discovering the cycle-time traps in your current development process and using cycle-time technologies to overcome these time sinks. For example, the fuzzy front end of the development cycle typically consumes half of the total time, which presents huge opportunities to save time. However, few companies think of using rapid prototypes here, even before the normal concept phase.

Astute application of RP technologies to overcome specific bottlenecks in the development process often saves much more cycle time than the hours or days that faster model making would suggest. Our goal here is to discover these opportunities that can really leverage the speed of RP and turn them into a business advantage. To the extent that we can clearly provide these business benefits, it becomes much easier to get approval from management for acquiring and using RP systems and services.

Where Does the Time Go?

Those making rapid prototypes are fond of explaining that the prototype they are showing you was made in just a few hours, whereas it would have taken several weeks doing it the old way, a time saving of 95 percent or better. Pretty impressive! Unfortunately, this is the technologist's view of the world. The business manager's view of the same situation is that the total development schedule for this product is two years, and the rapid prototype's saving of several weeks only amounts to a few percent of that two years. Not so impressive anymore.

Worse for the rapid prototypes, management is unlikely to be very supportive of RP when the savings appear so meager in management's terms.

However, there is a way to get much more impressive savings out of rapid prototypes, even when measured by the manager's stiff criterion. Fortunately, there are portions of the development cycle where months slip by unnoticed or months get wasted. If we can discover these opportunities and use RP to overcome them, then we have received great leverage on the weeks saved by making the prototype rapidly rather than traditionally. In other words, if we limit ourselves to the direct timesavings of RP, we are forfeiting its real potential to compress development schedules.

As examples of the potential, we will examine two parts of the development process that typically offer great cycle-time savings, the fuzzy front end and the transition to manufacturing.

The Fuzzy Front End

Although many companies have reduced their development cycle time greatly over the past several years, most of them conveniently choose to ignore the biggest opportunity to get products to market sooner. The portion they ignore is the one before they get started, what we call the fuzzy front end. They define this period out of existence, simply by saying that the project starts when they approve or staff it. Unfortunately, customers or the marketplace do not care about this invisible internal milestone, and only judge the product on when it is available for sale. Fast release to the market place depends on the length of the development cycle and also on when this cycle starts, that is, the length of the fuzzy front end.

This wouldn't be an issue if companies were quick to jump on a product opportunity and start developing it. But most companies take as long to start a project as they do to develop the product. This is illustrated in Figure 1 for three projects from one of our clients. Each of the three timelines has three significant milestones:

- When the need for the product is first apparent in the marketplace.
- When the company fully staffs a development team.
- When they ship the first unit to a customer.

For the three projects illustrated, the client spent longer on two of them getting ready to start the project than they did in actually developing it. Furthermore, these time splits are fairly typical. The fuzzy front end is indeed a major opportunity to compress cycle time.

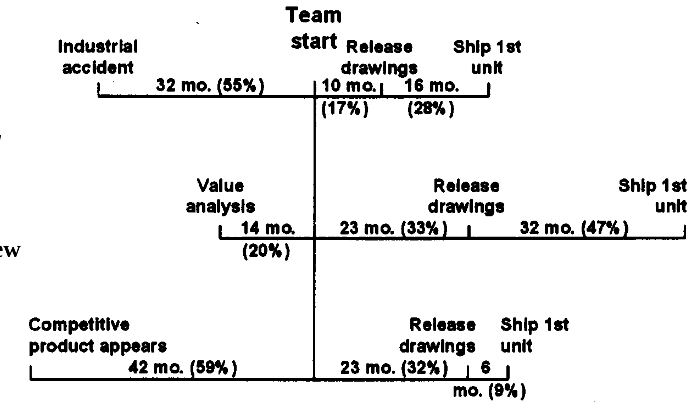

Figure 1. The fuzzy front end often offers larger time-saving opportunities than the remainder of the development cycle. Used with permission from **Developing Products in Half the Time: New Rules, New Tools,** *Preston G. Smith and Donald G. Reinertsen, John Wiley & Sons, 1998.*

Our job is to discover how rapid prototypes can have a major impact in the fuzzy front end. To do this we must analyse the time spent in the fuzzy front end for a few projects, constantly considering opportunities for applying RP. For example, huge time periods often slip by because critical decisions are not taken. Further investigation may reveal that the decisions fail to occur because certain underlying issues are not well defined. These issues may well involve alternative design concepts to suit different customer requirements. So, one way to "force" a decision is to create models for alternative design concepts, circulate them among decision makers, and encourage them to make a choice. Note that we have now saved the months that the project may float along awaiting a concept that emerges as best. Also note that the prototypes needed to force the decision usually need not be fancy or highly accurate ones. In fact, a few pieces of existing or competitive products, cemented together, may be adequate. At this stage a highly precise RP system is overkill.

Engineering Release

Another place to find cycle-time opportunities is the decision point between engineering and manufacturing. Because the commitment to enter manufacturing usually involves considerable capital to procure tooling, this is a major decision point. What often happens is that, because there is big money involved now, the decision-makers—upper management and marketing—really get serious. As they rethink the product, they sometimes ask for changes, which, of course, means delay. Assuming that the product meets its hard engineering specifications, these changes usually occur in softer areas, such as shape, feel, and appearance. This is just where rapid prototypes excel: in getting agreement on such soft issues.

"Management and the owners of the business, who make the ultimate decisions on RP technologies, are far more interested in getting a new product to market quickly than they are in the burst speed of a particular technology."

For example, we once sat through an engineering release review of an impressive electronic product employing sophisticated digital signal processing and housed in a small plastic case. The go–no-go decision was basically up to the Director of Marketing, and she reached her decision based not on how well the device could detect signals but on how it looked and felt. Fortunately, the development team, in this instance, knew that case appearance was crucial and had done considerable research on its shape and size. But they had also done a great deal to keep the marketing director in the loop from the beginning by showing her rapid prototypes of design concepts and getting her commitment—not just her reaction—to the design direction.

Thus, the two keys to using rapid prototypes to avoid reversals later are:

• start showing rapid prototypes to the final decision makers early, so you can incorporate their subjective desires into the design before it costs you redesign time
• make it clear to these decision makers that you need not only their reaction to the design but their early commitment to it based on these prototypes, because it will delay the project to "get serious" at a review later.

Finding the Time-Saving Opportunities

Above we have illustrated just two of the more likely places to find cycle-time opportunities that RP can exploit. While it is beneficial to have such candidates in mind in advance, it is also the case that these opportunities are highly individualistic. That is, they vary from company to company and even from project to project within a company. The bad news here is that you will have to do some work to find your particular opportunities. However, the good news is that, once found, your discoveries are unlikely to be of value to competitors—in management jargon, they provide you with a sustainable competitive advantage.

Below is a general process that will guide you in discovering your cycle-time opportunities.

Understand the Business Drivers

First, understand the role of new products and their development on the success of your business. Management needs a certain number of new products, and if these products could be developed for nearly nothing, management would be happy. In some companies, this is the extent of management's interest in new products. Rather than being primarily concerned with new products, the company's strategy may be more oriented toward controlling the distribution channel or servicing and supplying consumables for products that have been in the field for years. It will be difficult to get management's interest in any time-compression technologies for product development if management does not believe that new products are critical to the success of the company.

Next, understand the company's strategy for new products. Does your company want to be a leader or a fast follower, letting others take the larger risk in pioneering? Does it want to be known primarily as a leader in innovation, in cost, or in reliability and quality? Does it want to get to market as fast as possible, or does it really want to meet a specific launch date, such as availability for a trade show?

Each of these choices will determine the kinds of time-compression solutions you will seek. For example, to improve schedule predictability, you will take certain steps to minimise schedule risk, and you will procure technologies toward this end. However, because these steps will add time to the schedule in most cases, they are not wise for those who primarily want to get to market as fast as possible, so this choice suggests different types of solutions. As this example illustrates, if you cannot make the kinds of distinctions mentioned in the previous paragraph, you will be limited to generic solutions. These are not very powerful, nor do they give you a competitive advantage.

Finally, understand, for your products, the relative importance of the four project objectives: schedule, project expense, product unit cost, and product performance. Reference 1 (Chapter 2) explains how to make these calculations. Time is not always the most important driver of business success in a development project, and if this is the case for your company, you should be aware of it before you start looking for cycle-time opportunities. For your products, "rapid" prototypes may actually be more valuable for improving product performance than for actually being rapid.

This relative importance will vary by project, but you will probably be able to consolidate your development projects into two or three categories. For instance, one company, a motor vehicle manufacturer, found that they had some products that were new offerings for them. Time-to-market was the most important driver for these projects. Other projects developed improved models to replace models they were already manufacturing. For these, unit manufacturing cost was more important, because they could continue to sell the old model if the new one was late.

Know Where the Time Slips Away

Some technologies offer remarkable degrees of time compression. For example, a rapid prototype might be built in a day, whereas it took a fortnight to build it with the previous method, an impressive 90 percent reduction. However, there might be an opportunity to save three months out of the fuzzy front end of the project for less than the rapid prototype's cost. Or, for the cost of the rapid prototype, you might be able to modify a database to cut one day from each engineering change approval, which, for the hundreds of

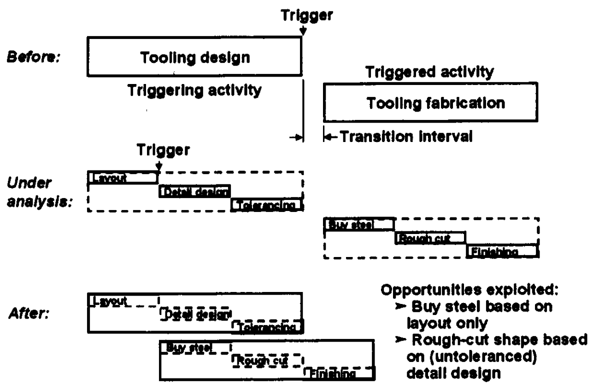

Figure 2. Discovering overlapping opportunities by identifying triggers within activities. Used with permission from* Developing Products in Half the Time: New Rules, New Tools, *Preston G. Smith and Donald G. Reinertsen, John Wiley & Sons, 1998.

engineering changes in a typical product development, could contribute far greater time savings than the rapid prototype would provide. Although the context of this paper is RP, other time-compression technologies may be more appropriate solutions for you cycle-time opportunities — don't overlook them.

There is no substitute for analysis here. Analyse several completed projects to see where the time went. Were there typically delays waiting for a decision? What would have helped to reach the decision faster? Was time wasted in redesigning parts? If so, did the designs have to be redone because they were done poorly the first time, which might suggest better engineering design tools or more training in using existing methods? Or did redesigning occur because marketing kept changing their interpretation of what the market wanted? This might suggest other capabilities, such as the ability to make concept models to clarify concept distinctions in customers' minds early. Unfortunately, many companies are slow to market simply because management tries to work on too many development projects at once, which dilutes the resources of all types on every project and stretches all projects out proportionately. Advanced technologies to compress time are likely to offer little benefit here. For instance, you might be able to make a prototype part in one day instead of ten, but if the manufacturing engineer, who is supposed to take the prototype to a supplier, is tied up on another project for the next ten days, the rapid prototype will provide no advantage.

This analysis of your projects provides an understanding of where the major opportunities are to save time in your schedule. Armed with this information, you become a far more savvy shopper for time compression technologies. Rather than being overly influenced by the technical wizardry being sold, you will know which solutions are likely to yield benefit and which will not in your current situation. You will also see how to. apply time-compression technologies to get more out of them than the face value of their time savings.

Redesign the Process

The next step, actually redesigning the process, is highly individualistic, so we cannot outline a process here. But we can offer some tools to help you: concentrating on the critical path, watching for the triggers, asking how fast could it be done, and thinking of time as a trade-off.

Concentrate on the Critical Path

The concept of phases, gates, or checkpoints seems to be deeply etched into development managers' minds. However, among the managers we know who do well at accelerating projects, another concept is at least as dominant: the critical path. The importance of the critical path is not its ability to be calculated by project management software and portrayed on a schedule network. Instead, its real value stems directly from its definition: any activity on the critical path that slips by one day will directly cause the project end date to slip by one day.

Experts at rapid product development are constantly aware of which activities are on the critical path, and they actively work to remove activities from the critical path. The critical path is the hot seat; it is not a pleasant place to be. We can keep activities off the critical path by using technologies, such as RP, to accelerate them. We can also keep them off the critical path by assigning more resources to them, completing them before they get onto the critical path, removing or relaxing the requirement for them (making them not critical to project completion), or rearranging project activities to allow more time for them. Critical path management is a juggling act: when something is removed from the critical path, something else replaces it. Such is the life of a fast-track product developer.

> *"Unfortunately, many companies are slow to market simply because management tries to work on too many development projects at once, which dilutes the resources of all types on every project and stretches all projects out proportionately."*

We find a critical-path mentality to be an extraordinarily valuable one for anyone to have who is connected with an accelerated development project. But this concept is also quite helpful to those redesigning development processes, because only activities on the critical path will directly influence the project end date. This greatly narrows our search for the most fruitful means for compressing the schedule.

Watch for the Triggers

If we look at a Gantt chart, as in Figure 2, it may appear that activities cannot be overlapped. For example, we must build a prototype before we can test it; a part must be designed before it can be built. But if we dissect these activities, by looking inside them on the Gantt chart, we usually find that part of the succeeding activity can be started before its predecessor is complete.

The concept of triggers is helpful here, as shown in Figure 2, which illustrates the design and building of tooling. The trigger for starting the tool-building activity is buying the piece of tool steel. This trigger occurs not when the tool's design is complete, but much earlier when overall dimensions of the tool are available.

Once we recognize that the tool-building trigger occurs relatively early, we can overlap many of the design and building activities. And once we find an early trigger for the successor activity, we can reassess both activities to optimize the flow of their subactivities so that they can be run simultaneously. Discovering an early trigger thus becomes the starting point for designing overlapped activities. The key to finding such triggers is to probe into the internal structure of larger activities such as tooling design and tooling fabrication. As long as we see them as one big activity we will never find the opportunities inside for overlap.

Clearly the quality of our information plays importantly here. If the length of the tool increases after we have bought the steel, we have a problem. However, if we are not completely sure about the length of the tool, we might want to pay more for the longest piece of tool steel we might need, because the time value of having the steel early might outweigh the extra cost of buying a bigger piece.

How Fast Could It Be Done?

Sometimes we can break out of old patterns by freely considering how fast something could really be done. For example, consider a related situation: writing, editing, and publishing a book. Using a conventional approach, such a project takes twelve to eighteen months. But fast-cycle publishers, such as the publishers of the popular Dummies books on personal computers and other subjects, can produce a book in just three months, from idea to store shelves.

This represents a 4–6X acceleration, and it is not the limit. Books on major events, such as Princess Diana's life and death, appear much faster. Or compare a book with a daily newspaper, which is about the same number of words and requires similar development and manufacturing steps. This gets us down to a matter of hours, from a project that traditionally has been considered to take a year or more.

Such an unconstrained approach can expose large opportunities for overlapping. For the moment, do not let money, people, or other resources stand in the way. Try to find a similar situation that was done much more quickly. If you let yourself go here, your speedy alternative is likely to be rather outlandish. But now you have something against which you can start comparing real-world solutions. For example, if a newspaper composes half an edition before the news comes in, why can't we buy half of our parts before we have a design? Maybe we should be relying more on standard parts, so we can devote more of our creative energy to the parts that "make news" (make our product more distinctive in the marketplace).

Alternatively, look for examples where something got done exceptionally fast, even if it was only once, in your own company. On one consulting assignment we were assisting a task force trying to cut part of their development process down from thirteen months to three. The task seemed hopeless to these old-timers, and we weren't making much progress. Then a respected member of the group noticed that, once before, they had done the job in only four months. By studying this example, they soon had a plan for routinely doing the job in four months. Mathematicians often do their work by starting with an "existence proof," first demonstrating that there is in fact a solution without worrying yet about its nature. Sometimes, some of the rest of us need an existence proof to spur us into finding a solution.

Think of Time as a Trade-off

Earlier it was mentioned that product development projects normally have four objectives:

- Develop a product with a certain set of features and with certain performance levels, as listed in the product specification (performance objective).

- Satisfy a target unit manufacturing cost for the resulting product (cost objective).

- Do this within a certain development budget for the project (expense objective).

- Complete the project within a given time (schedule objective).

It was suggested that you should understand the relative importance of these objectives for your projects, because this will influence your strategy for employing rapid prototypes.

However, there is also a tactical reason for knowing quantitatively how these objectives are balanced. The balance points allow you to trade off one objective against another. To effectively compress our development cycles, we must constantly consider time as an objective to be traded off against the other objectives.

This may seem obvious, but as managers, we are taught to manage each of these objectives—performance, cost, expense, and schedule—to its target independently and to control variances of each one. In so doing, we lose sight of the fact that some of these objectives may be far more important than others. If we could cut a week out of the schedule, we would probably be pleased to pay £100 for it. But if the cost of saving a week rises to £1,000,000, we might decide that the week is not that important. Just how important is a week of schedule compared to development expense? How much schedule time would I be willing to give up to cut unit manufacturing cost by £100? How much delay should I accept to regain a loss of 10 percent in product performance?

To really leverage cycle-time technologies, we need all of the flexibility we can get, and these trade-offs give us a great deal of flexibility. The four objectives are in fact a balancing act. We actively seek opportunities to trade off one against the other, gaining net advantage. We can always find more expense money to buy some time at a bargain price. We will refuse to delay a project to add another product feature if the price (in time) is too dear.

Changing Habits

If you are satisfied with the direct benefits of perhaps a few months that rapid prototypes can bring you, then your life will be relatively simple. You can procure an appropriate RP system, train a few people in its use, and proceed to enjoy its relatively meager benefits. However, if you wish to take advantage of the much greater leveraging benefits offered here, there is one more step, the hardest one of all. Throughout the organization, people will have to change their habits.

For instance, it was suggested in the beginning that great amounts of cycle time could be saved by getting certain decision makers to commit to a concept early on the basis of a relatively crude model. Experience has shown that that these decision makers are reluctant to make commitments on the basis of a model—and it matters little whether the model is crude or refined.

Decision makers are used to seeing the real thing; anything less is a real change for them. Yet, we have seen such executives make the change and become comfortable with such early commitments. This generally requires two things:

• They must see the benefit to be derived from their change.

• They must become comfortable with the new process and the people executing it, so that they can trust it.

The benefit should be straightforward to describe after you have mapped out the original and the compressed processes and have calculated the trade-off value of cycle time for your projects. The comfort and trust part depends on changes required of both the developers and the executives. The developers must be more open with early information, and the executive can practice more MBWA.

The choice is ours. We are at the threshold of a new era in RP. The fuzzy front end offers enormous business potential, and the conceptual modelers just appearing are supposedly aimed at this opportunity. The conceptual modelers themselves are much cheaper than their full-fledged cousins are. Will the models coming out of conceptual modelers be considered as “toys,” or will they lead to unprecedented breakthroughs in total development cycle time?

References

1. **Smith, Preston G., and Donald G. Reinertsen**. *Developing Products in Half the Time: New Rules, New Tools.* John Wiley & Sons, 1998.

Preston G. Smith has concentrated on helping companies to accelerate their product development since 1984, both as an internal staff consultant and as an independent Certified Management Consultant.

His client base spans well over a hundred companies in 17 countries. He has 20 years of prior engineering and management experience, holds a Ph.D. in engineering from Stanford University, and is co-author of Developing Products in Half the Time: New Rules, New Tools.

Author's note: This article is reprinted here in Portable Document Format exactly as it appeared originally (sans advertisements that appear on the original pages). However, the two figures were small originally and somewhat less sharp in PDF format. Consequently, they are reproduced below for the benefit of your eyes.