From Experience:
Reaping Benefit from Speed to Market

Preston G. Smith

Many companies have gained great benefit from a speed-to-market program. Others have tried but failed to achieve measurable, sustained value. Still others have not even tried, being concerned about adverse consequences.

This article helps newcomers decide how to approach time to market. It first addresses some common fears about development acceleration, such as increased project expense and lapses in quality. An effective program overcomes these concerns, but it must do much more. It must stem from a statement, clear to all developers, explaining how faster development is tied directly to improved competitiveness and thus to higher profits for the company. A general corporate goal of a 50% across-the-board cut in cycle time will not do, nor will an underlying hope that faster development will improve developer productivity.

To help establish a clear link from speed-to-market to profitability, I suggest calculating how much a week of delay impacts profit, and I show how to align a development acceleration program with corporate programs to improve quality or productivity. © 1999 Elsevier Science Inc.

Introduction

Over the past decade, time to market has moved from obscurity to a prominent topic among product developers. Many manufacturers have cut their development cycles in half or better—or so they claim. Although many, including Black & Decker, Motorola, and Chrysler have made impressive gains, many others have not been able to keep it up. All too often, a company places great emphasis on one project, gets its glamour story written up in a prominent trade magazine, then falls back to its former ways.

I have been involved in many of these cycle-compression projects. The successful, permanent improvements have been gratifying to all of us, but we have been frustrated by the flash-in-the-pan performances. The purpose of this article is to apply our “From

BIOGRAPHICAL SKETCH

Preston Smith has specialized in shortening manufacturers’ product development cycles since 1984, initially as an internal consultant and for more than a decade as head of New Product Dynamics, a consultancy and training firm concentrating on time to market. His clients number over a hundred and range from food companies to computer, medical device, and telecommunications manufacturers in Europe, North America, and Asia. He is co-author of *Developing Products in Half the Time,* a leading book on cycle compression, and has published many articles on this subject. He is a popular speaker on development acceleration issues at professional conferences and corporate meetings. Prior to his concentration on cycle time, Preston gained 20 years of experience in engineering and management at several firms. He holds a Ph.D. in engineering from Stanford, is a member of PDMA, and is a Certified Management Consultant.

Address correspondence to Preston G. Smith, Certified Management Consultant, New Product Dynamics, 3493 NW Thurman Street, Portland, OR 97210. E-mail: preston@NewProductDynamics.com.

J PROD INNOV MANAG 1999;16:222–230

655 Avenue of the Americas, New York, NY 10010

0737-6782/99/$–see front matter
PII S0737-6782(98)00059-9

Experience” to illuminate the differences in these two kinds of performances and draw conclusions as to what is needed to obtain measurable, sustained benefit.

I would prefer to concentrate on describing these keys to success so that managers could just implement them. Unfortunately, there are many pitfalls along this road to success, and many managers are lured into them. Therefore, I start by outlining the pitfalls so that you can avoid them—and so that you are aware that the route to success is not as straightforward as it might appear.

Unproductive Routes

Many companies get off to a weak start because of unaddressed concerns that some of their people have about what time to market means or how it will be approached (Exhibit 1). Let’s see what some of these concerns are, where there is some truth to them, and how they get us into difficulty.

Skip Steps to Save Time

Some observers worry that acceleration will be accomplished by simply skipping some of the work to be done, especially in understanding the customer and in testing prototypes. Although skipping steps does happen, intentionally skipping vital activities is simply not a viable option these days. Most companies have made huge gains in product quality in recent years, and it would be foolhardy to reverse these gains. Customers now expect higher quality products, and the legal consequences of shipping half-developed products can be horrendous.

I have seen cases where developers have skipped steps to save time, with catastrophic consequences. These cases have usually occurred when management was unclear about its cycle-time objectives, which left developers to conclude that skipping steps was an acceptable option.

Yet, there are often some steps of a firm’s development process that do not add commensurate value. These should be candidates for removal. For example, design automation tools have made certain checks on drawings and designs superfluous. Or if we choose to reuse older components in a design, then certain checks and tests may not be needed. To slavishly stick to completing all of the steps wastes labor and time. Thus, a certain amount of flexibility is advisable, and the faster companies generally have such flexibility built into their development processes.

Exhibit 1. Concerns About Going Faster

Countless articles have appeared in the business press featuring time-to-market success stories [6–8]. Not so apparent are several articles [2,4,5,9,14] and books [1,15], mostly by academics, that tend to be more cautionary and tentative on the wisdom of time compression. Although this set of more cautious literature is not as voluminous as the success stories, I take it seriously. If poorly done, time compression can lead to errors. Furthermore, I believe that these issues also voice the unspoken concerns of some managers in industry who would like to get their new products to market faster but feel that the side effects may be unacceptable. This literature describes the side effects in various ways[1]:

- “Many mistakes happen when skipping steps sacrifices necessary information” [21.
- “Speed is everything! But not at the expense of quality of execution” [1].
- “In general, there may exist a tradeoff between speed and design quality” [9].
- “In essence, time is bought with increased development cost” [5].
- “Once embarking on the route of shortening life cycles, it becomes increasingly painful to stop and increasingly costly to continue” [15].

Clearly, in order to enjoy any benefit from speed to market, we will not only have to avoid these side effects but also address the concerns underlying them. The accompanying article indicates how one can enjoy the benefits of speed to market while overcoming the potential negative side effects.

[1] I would argue that some of these conclusions depend on inadequate mathematical models, thus they are questionable. However, the point of listing these quotations is not to dispute their validity but to indicate the authors’ underlying concerns.

A variation on this theme is to skip certain product features when time turns out to be more precious than the feature. Done properly, such skipping is one of the tools of rapid development. However, skipping features is not a casual or hidden event; it should be done by the cross-functional development team and be supported by decision rules applicable to the project. (I discuss these decision rules later.)

In reality, even when management’s objectives are clear, some activities can get compromised due to pressure to get the product out. For example, one client in the heavy equipment industry is experiencing rising warranty expenses as it compresses its development cycle. This highly regarded firm is not taking this setback lightly, but it is not letting it stymie progress either.

In conclusion, skipping steps is a valid concern, and it does happen. It can be combatted through clarity both in objectives and in what can be given up in the tradeoffs (a do-it-all directive gives developers little help in making such distinctions).

On the other hand, a company that never "skips steps" is probably not aggressive enough. The solution here is not in religiously completing all of the steps but in an effective process review and improvement procedure that keeps major mistakes from being repeated [11].

Speed Is Too Expensive

Many managers are concerned about the expense of compressing cycle time, and Graves [5] elucidates such a tradeoff between time and development expense. He suggests that this tradeoff is inevitable, but our experience with clients shows that expense can both rise and fall with cycle compression. For instance, Figure 1 illustrates a favorable correlation for the major automobile manufacturers in the United States for comparable projects.

Although it would be nice if expenses did not rise with faster development, rising expenses are actually a blessing in disguise, because they provide us with a powerful cycle-compression tool to "buy" time. Later I show how one can assess the cost of delay for a project and use it to accelerate development at bargain rates. Likewise, when speed is no longer a bargain, we cease "buying" it. However, most companies can gain a great deal of cycle time before the cost of the time saved becomes too dear.

As I explain later, there is usually an "expense" to be paid for faster development, but it is broader than just expense dollars. For example, it is often paid in labor to improve the development system[2] or in the agony of changing familiar behaviors.

Speed Really Yields Productivity

In these lean times, many executives seek cycle time not for its own value but as a means to raise productivity by squeezing more products out of their resources. Their reasoning goes, "If we can develop a product twice as fast, then we can complete twice as many this year." They fail to accept that faster development requires enhanced staffing levels (over shorter time intervals). Without the enhanced staffing, nothing changes, and their developers soon become frustrated by the implied demand to work twice as hard. If the desired objective is actually productivity (more new

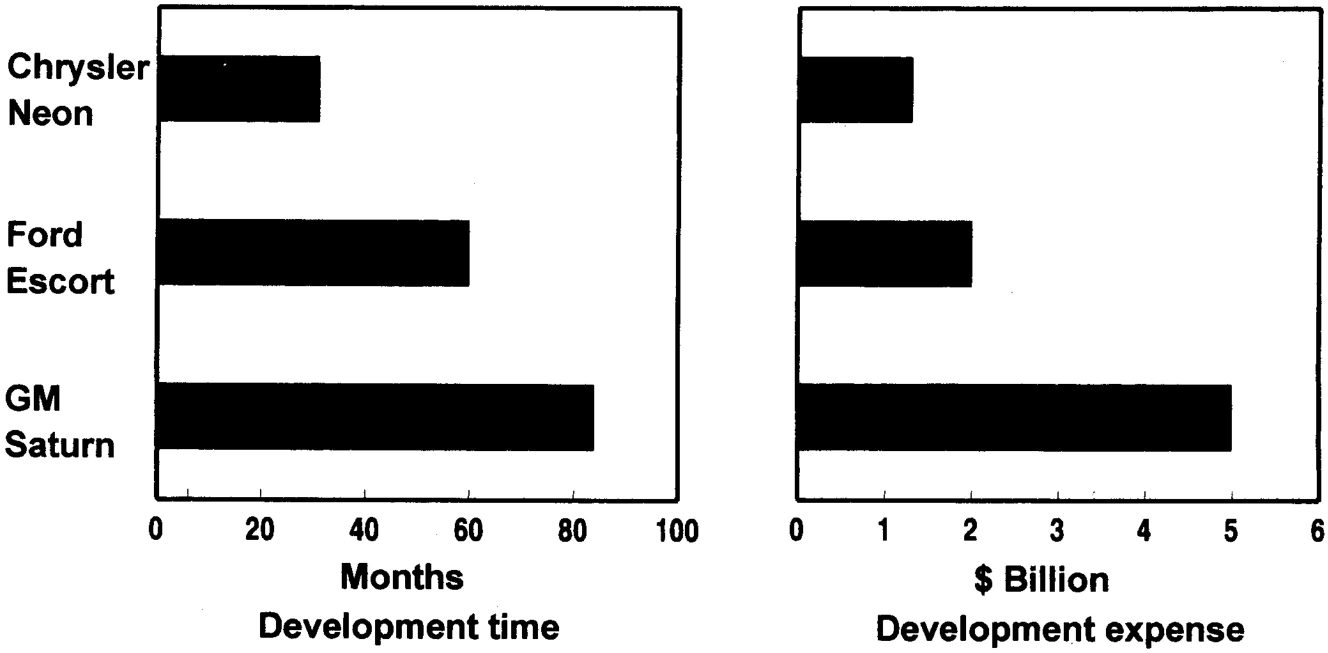

Figure 1. Comparison of similar programs for three automotive companies shows that time and expense need not be a tradeoff. Sources: *Ward's Auto World,* November 1993, pp. 38ff; *Design Management Journal,* Summer 1995, p. 50.

[2] *Development process* is more common terminology than *development system.* However, superior product development involves more than just the process used. For instance, adequate staffing levels, appropriate skills, and effective enabling technologies also are critical. *System* is an attempt to encompass these factors too.

products per developer) rather than cycle time, then other solutions, such as design reuse or design automation, are likely to be more effective than simply trying to optimistically extract twice as much output from the same resources.

Although companies, such as Chrysler in Figure 1, have gained in both cycle time and productivity by concentrating heavily on cycle time, concentrating on productivity under the wraps of cycle time is likely to backfire. First, when the troops discover that cycle time is just a guise for getting them to work harder, they are likely to revolt. Second, focusing primarily on productivity is likely to starve the system of the resources needed for change just when the system needs mammoth change. To achieve its results, Chrysler spent over $1 billion on a new development center that allows them to co-locate their development teams, a potent tool to accelerate decision-making. Chrysler also spent generously on improved computer-aided design (CAD) systems, team training, and supplier development. Keep in mind that Chrysler made these commitments around 1990, when it had no cash to spare.

The Price of Accelerated Development

Although many companies have made dramatic improvements in time to market—often cutting their development cycle to less than half of its original duration—I do not know of one instance where the transition has been simple or painless. Consider what Neil Hagglund of Motorola has to say about their cycle-time journey:

> First, in my more than thirty years of product development experience, working with some of the best product developers in the world, I have yet to find a single magic tool for transforming a development process. Other companies may jump from fad to fad hoping there is a fast, easy way to accelerate product development. At Motorola we achieve rapid development the same way we achieved breakthroughs in quality—with old-fashioned hard work and constant management attention....
>
> Finally, I would encourage you to stay the course on this effort. The benefits of faster development can be substantial but they cannot be achieved instantly. Fundamental changes in your development process require careful analysis, broad involvement, and extensive effort. Not everything that is worth changing can be changed quickly. If you approach this as a short, quick journey you will not get very far [13].

Along with Motorola and many other successes, I have seen too many instances where executives have failed to achieve measurable reductions in cycle time. In almost every case this has resulted from their illusion that it would be simple and quick. In a typical failure, an executive asks us to provide a day or two of training or to address a corporate task force considering “faster, better” product development. In the absence of clear objectives and substantial follow-on work, this minimal dose is simply inadequate to make a dent in their complex system.

From the successes, such as Motorola, and from the failures, I have concluded that considering rapid development to be free or easy is a prescription for failure.

Companies that make truly impressive reductions in cycle time generally gain time by paying for it in the two ways described in the following sections: ongoing tradeoffs against other development objectives, and in the cost of making the organizational transformation required.

Tradeoffs Against Other Development Objectives

Most development projects have four objectives: schedule time, development project expense, unit manufacturing cost of the resulting product, and product performance (including the product feature set). Although management is fond of saying that they want it all, in reality, product developers trade off these four objectives against each other as they work.

Often, these tradeoffs are so automatic with developers that they do not think of them consciously. They make the decisions subconsciously using guidelines they have assimilated over time. For example, if management has questioned the expense of early rapid prototypes that revealed design mistakes, then developers will defer making any rapid prototypes until they are certain that the design is perfect. Consequently, they waste time double-checking the design to ensure that they do not waste prototyping dollars.

Notwithstanding the “do-it-all” directive, developers must make such decisions as they work. Thus, in reality, when developers emphasize time, the other three objectives tend to receive less attention as these numerous decisions accumulate.

Once we recognize that such tradeoffs occur spontaneously and subconsciously, the opportunity for cycle-time management is to bring this tradeoff phenomenon into the open. As described later, quantitative tradeoff rules can be developed for each project. Then developers can explicitly assess each decision they make to ensure that time receives the desired emphasis relative to other project objectives.

In short, effective managers of cycle time know that gaining time means giving up some of another project objective, and they analyze each project decision to obtain the mix of objectives that will maximize profit.

Costs of Organizational Change

In addition to the tradeoffs, there are costs for initiating and sustaining the organizational change required to adopt a new approach, as expressed earlier by Mr. Hagglund. Motorola has invested heavily in the changes it has made, and it is justifiably proud of them. Chrysler has also spent generously for its envious success, as indicated earlier.

These substantial costs have a couple of implications. One is that unless the organization is ready, able, and willing to make this kind of commitment, they are most likely wasting their resources by paying lip service to rapid development. As Robert Waterman put it in the original edition of our book, “Don’t bother to read the book unless you, those around you, and your top management team will treat this subject as more than the latest management fad. Save your money and get your vicarious thrills at the movies instead” [12].

The second implication is related. Such major operational changes are quite unlikely to occur unless upper management leads them. One partial solution here—followed by many companies, sadly—is to relegate product development, and thus rapid development, to one department, usually engineering, where it can be addressed at a lower level. Of course, this is just the old compartmentalization of product development that we have all been trying to overcome in recent years. But this does make the job easier. Engineering alone can now procure and use the latest CAD systems and train themselves to do a wonderful job on failure modes and effects analysis (FMEA), robust design techniques, or value engineering. Unfortunately, the biggest schedule-compression opportunities happen to be in cross-functional areas, such as jointly developed product specifications, the fuzzy front end, and the transition to manufacturing. So firms that implicitly elect to view the issue in limited terms are really electing to obtain quite limited returns.

In our experience, these organizational changes in behavior are often most difficult in the early phases of a project. In contrast, the tail end—the transition to manufacturing—normally occurs on a tight schedule anyway, because as initial milestones slip, the later participants are somehow expected to make up for time lost earlier. In most cases, manufacturing welcomes better attention to schedules, as this gets them away from always receiving the brunt of the schedule crunch.

In contrast, the people involved at the front end are more used to taking the time to “do it right.” They tend to believe that customer preferences need time to emerge, that product concepts must have a while to ferment, and that many constituencies must have time to “think about it,” reaching consensus before moving ahead in an unwise direction. For example, the fears about accelerating development, quoted in Exhibit 1, mostly come from those most heavily involved in the front-end planning and marketing activities.

I mention this only to suggest that it is the participants in these early activities, for example, those in marketing, who often will be faced with the greatest changes in attitudes and behavior as a firm moves to shorter development cycles. Yet, it is precisely at the beginning of a project, in what we call the fuzzy front end [13], where the greatest opportunities usually lie to compress schedules. By ignoring the time-compression opportunities in the fuzzy front end, which may demand the largest changes in habits, we also forfeit major benefits.

Putting Accelerated Development on a Profit Foundation

This is where we shift from considering the sidetracks that distract us from rapid development to covering the essentials of a successful development acceleration program.

Most organizations developing new products are for-profit organizations, and their prime objective in developing products is to make a profit. Consequently, there is no better way to become and remain focused on the real benefits of development acceleration than to place it on a profit foundation. If faster development earns us more profit than alternative uses of our resources and energy, we do it; otherwise, we don’t. To make knowledgeable decisions on time to market, it must be related to profit. Calculations for doing this are outlined in Exhibit 2, and the resulting factor for converting delay into profit is called the cost of delay.

Base Decisions on the Cost of Delay

Once we know the cost of delay for a project, we can and should base all proposals to compress time on this factor. Laboratory technicians and CEOs alike can employ this same cost of delay factor. This ensures that time-to-market decisions are made consistently across the organization. Moreover, because such decisions can be made at the developer level, they can be made faster and with confidence that the decision reached by the developer is the same one that a senior executive would have made. Thus, use of the cost of delay speeds up decisions and empowers the project team to proceed without waiting for approval or wasting effort in being overruled from above.

Exhibit 2. Calculating the Cost of Delay

The cost of delay is a value that tells you how much profit—on a before-tax basis—you will lose if the product is delayed by a day or a month, whichever period you prefer to use.

The calculation is a straightforward sensitivity analysis. First create a profit-and-loss statement for the life of the product, including its development period and its sales life. This normally can be done on one sheet of paper using spreadsheet software. Keep the model simple, because a simple model will facilitate buy-in, and this is all that is needed or can be justified by the accuracy of the data used.

This is called the baseline model, because it assumes that the project goes well: on time, on budget, etc. Now think about what would happen if the product were 6 months late. How many orders would you lose? How many of these might be regained later? How much market share might you lose permanently because a competitor gained a stronger foothold? Would you miss a premium pricing opportunity or have to lower your price?

Express this late scenario as a variation of the baseline spreadsheet. Now subtract the cumulative lifetime profit of this lateness spreadsheet from the corresponding profit of the baseline spreadsheet to get the profit lost due to lateness. Divide this number down to obtain the cost of delay in the terms you desire. For example, if you prefer to express the cost of delay on a daily basis, divide by 180.

The cost of delay is expressed on a pre-tax profit (not revenue) basis because it is this kind of money that we can use to "buy" time-saving opportunities. You also can calculate other useful decision rules, for example, the profit impact of missing a product feature or having a higher unit manufacturing cost. For additional information and help on building the model, checking it, keeping it simple, and gaining ownership of the results, see [13, Chapter 2].

I am not trying to give time compression special treatment here. I am just proposing that it be subjected to the same hard-nosed, cost-to-benefit process that would be used routinely for a new machine tool in the shop. If we do not apply this type of explicit cost-to-benefit analysis, the normal course of events is usually a quite subjective approach to evaluating time. The problem here is that, because time has less tangible impact than other factors in the tradeoff decision, such as development expense, we tend to undervalue time, usually greatly.

It would also be erroneous to conclude that we must always pay dearly for time. Often, when we run the numbers, we find that time can be bought at remarkably beneficial rates. Sometimes, as shown for the three automobile makers in Figure 1, time can be gained while other tradeoff objectives also become more beneficial. For example, a true cross-functional development team often leads to advantages in all objectives at once. However, if it were this simple, the rational manager would already be taking the time-compression path indicated by the tradeoff analysis. The difficulty here is that there is still a price to be paid in organizational change, so some managers choose not to pursue opportunities that the financial analysis shows would benefit them in multiple ways. It simply seems too difficult to make the organizational change needed to reap the financial benefit.

Expect the Cost of Delay to Vary Widely

The cost of delay varies by factors of ten to more than a thousand, even for companies in the same industry—even for different projects within the same company. It follows that there are no universal values, such as "six months of delay equals one-third of a product's lifetime profit."[3]

Even generalizations, such as time to market being more important in high-tech industries than in more mature ones, are risky. A high-tech market now actually may be competing on a basis other than cycle time, whereas a mature company may be able to gain substantial advantage over its direct competitors by being faster to deliver the goods than they are. For instance, one company producing electrical machinery found that it was missing out on market opportunities that it discovered at the industry annual trade show. It was too slow to introduce a new product in response at the next show a year later. So it aimed specifically at

[3] This value often is attributed to a so-called McKinsey study [3]. However, this analysis was actually done by Donald Reinertsen while he was at McKinsey & Co., and in the source article Reinertsen makes it quite clear that the cost of delay depends on the specific circumstances [10].

this target. Once it had its cycle down to under a year, it saw little point in further reduction and thus shifted its next improvement efforts to beating competitors on cost and quality too.

A similar faulty generalization is to attribute the importance of speed to innovation leaders. Ironically, pioneers often have the luxury of time, simply because they may have little competition until they announce their product. In contrast, the clock is clearly running for the follower to this pioneer.

In conclusion, there is no substitute for running the numbers for *your* projects. The cost of delay is likely to vary by a factor of at least ten for various projects that you have under development today. We have calculated costs of delay for some projects at only $2,000 per day, whereas others came out at $1,000,000 per day (in pretax profit). We even caught one team leader cutting his cost of delay in half, because he was afraid that senior management would not believe the true value!

A supplementary benefit from running the numbers for your projects is that, as you do it, you will start to see some patterns that may have not been so clear before. Delay is likely to be a far bigger factor in profitability for some types of products than for others. For instance, one client found that cycle time is more valuable for new additions to its product line than it is for model replacements. With this kind of information you will know where to put your time-to-market emphasis, rather than pursuing a bland cut-cycle-time-in-half policy across the board.

Use the Cost of Delay Both Strategically and Tactically

Once you have a quantitative appreciation for the cost of delay for a range of your products, use this information both strategically and tactically. The tactical application has already been explained: use the cost of delay, as well as the tradeoff rules among the other three project objectives (project expense, manufacturing cost, and product performance), to make daily project decisions related to "buying" time on the project.

Strategically, use the cost of delay for setting directions in a time-to-market program. Calculate the cost of delay for a representative assortment of your projects. Then calculate an aggregate value for the whole organization by weighting the cost of delay appropriately over your projects, for example, weighting them by revenue or expected profit contribution.

This aggregate cost of delay will be useful for deciding whether to undertake a time-compression effort at all, whether to invest in design automation technologies to help your engineers work faster, or whether it is worthwhile, for example, to co-locate development teams to speed up their activities. If, in the process of calculating costs of delay for your projects, you discover segments of your business that have quite different costs of delay than others, you will be able to focus your strategy better. For example, you could approve an advanced CAD system for one business segment but not for another.

Some clients have made profound changes in their product development systems based on insight from their aggregate cost of delay. One client, for example, found that the signature authority of their development team leaders was equivalent to only 4 hours of project delay. Because management wanted teams to be looking for savings of months instead of hours, top management totally revised the project budget approval process, basically giving the team full authority for its budget after initial project approval.

Implementation Alternatives

Because hard-pressed managers want to gain on all fronts at once, and because companies such as Chrysler have indeed made progress simultaneously on several fronts, what is the best set of objectives to pursue? For example, is a cycle-time program tantamount to improving productivity? Knowing your aggregate cost of delay and other decision rules will help you make these decisions. However, there are some other considerations too. Let's explore some of these implementation options.

Concentrate on Productivity

With today's lean corporate environments, a focus on productivity (number of new products per unit of resources expended) is attractive. As discussed earlier, this is what many managers pursuing a cycle-time program really want.

As mentioned before, you should be clear with yourself and with your people about the distinction between productivity and cycle time. If you are not, the program is likely to backfire and you will not achieve either objective.

Should you decide to work toward productivity rather than cycle time, remember that this objective, if it is to be truly successful, will also require substantial organizational change. In turn, organizational change will require resources. Thus, you will have to invest in a productivity program in order to obtain cost savings later. Just trying to cut expenses on existing projects will force you deeper into doing what you are doing now, perhaps alleviating a crunch in the short term but further entrenching unproductive methods in the long run.

Also, recognize that an emphasis on productivity is likely to stretch cycle times. One means of raising productivity is to ensure that no resources are ever sitting idle, thus guaranteeing queues. Queues are the antithesis of speed.

Concentrate on Quality

Many companies start with a corporate total quality program, then proceed from there with a time-to-market program as an extension of total quality. For example, this was the progression at Motorola, broadening from a Six-Sigma quality program to a 10X cycle-time program.

As stated at the outset, certain levels of quality are a given today, so we are not considering compromising quality to cut time, what is often referred to as skipping steps for speed. For the most part, quality and speed reinforce each other. For instance, a product specification developed by a cross-functional project team and well rooted in an understanding of the target customer is likely to enhance both quality and speed.

The main conflicts between quality and speed are likely to occur if quality is viewed narrowly as processes, procedures, and documentation to ensure that all steps are followed. This could be thought of as inspecting quality in rather than designing it into the development system. Such approaches, which check to see that all steps were completed satisfactorily, can indeed eliminate rework and thus time, but they also add time to the process. So their net effect on cycle time can swing in either direction. Another difficulty with such approaches is that they tend to be sized for your largest, most complex projects, so they overburden most projects with unnecessary steps.

Another warning is to avoid using staff people, for instance, those in a corporate quality function, to create development systems. Such individuals are more likely to create a lethargic system. They lack the first-hand experience of where the pitfalls really are, so they imagine unlikely problem areas. This adds time-absorbing baggage to your development process. Unfortunately, your most prolific product developers—your most valuable resources—are the ones best prepared to create systems for gaining speed and quality simultaneously.

I have found that companies having effective total quality programs in place have a distinct advantage in compressing their development cycles. The basic tools of total quality, such as affinity diagrams, effective meetings, brainstorming, and root-cause analysis, are just the implements needed to accelerate product development. Finding ways to take time out of systems comes quite naturally to those comfortable with such tools.

Concentrate on Time

Although a concentration on time, productivity, or quality will lead to many of the same improvements, a focus primarily on development cycle time can lead to the most comprehensively beneficial outcome. Cycle times can be short only if we employ resources effectively (high productivity). In addition, a fast system does not allow for unnecessary rework and redesign (poor quality). It is rather like a race car: in order to get around the track as quickly as possible, all systems must be working flawlessly. Time is perhaps the most comprehensive measure for those who really must have it all.

Conclusions

This article concentrates on speed as the parameter to be improved in product development. As discussed, speed is not always the correct objective. However, I suspect that many of the suggestions made here will, with some adaptation, be equally applicable to other objectives.

To sustain a speed-to-market program, you must know, in terms of the dynamics of your competitive arena, just how cycle time will translate to your bottom line. Unless management is able to explain this in an understandable way to everyone who touches a new product, the program will appear to be just another management fad. Worse, the program could get misinterpreted in harmful ways, such as skipping steps for speed.

Because speed is less tangible than other project objectives, such as development expense, speed is likely to be greatly undervalued unless you explicitly calculate the cost of delay. In short, time will lose out, because it is so slippery. Once you know the cost of delay, make sure that it is used throughout the organization to make development decisions. This puts time to market on a quantitative basis and empowers the development team to make quick, accurate decisions that will stick.

Of the many companies we have observed attempting to compress development cycle time, the primary difference between those that have benefited from a time-to-market program and those that have not is the extent to which they have considered it to be an investment in improving the business. As mentioned at the outset, those who regard rapid development as a means to get a few products to market quickly for immediate improvement in the bottom line generally get only that out of it. Those that limit their solutions to one department, such as engineering, receive limited and generally short-term benefits.

However, many companies are seeing product development—and rapid development—as a core capability of the business. It follows that this capability is worth investing in, just as management would invest in production capability by building a new manufacturing plant. These companies invest in hiring and training their developers. They invest in providing them with state-of-the-art CAD systems. Beyond these obvious investments, they invest effort in the development system itself. They invest in their development process, in getting better, faster input from customers and partnering with suppliers, and in the inner workings of their development teams. These firms continue to get faster and faster through continuous attention and continuous improvement [11]. In short, they sustain their dream for faster product development, and they know what it is doing for their bottom line.

I wish to thank Tom Hustad, whose suggestions on previous versions of the manuscript improved it greatly. Dennis Constantine and Gregg Tong also made valuable suggestions, which I appreciate.

References

1. Cooper, Robert G. *Winning at New Products: Accelerating the Process from Idea to Launch*, 2nd Edition. Reading, MA: Addison-Wesley, 1993.
2. Crawford, C. Merle. The hidden cost of accelerated product development. *Journal of Product Innovation Management* 9:188–199 (1992).
3. Dumaine, Brian. How managers can succeed through speed. *Fortune* 119:54–59 (1989).
4. Ellis. Lynn W. and Curtis, Carey C. Speedy R&D: How beneficial. *Research-Technology Management* 38:42–51(1995).
5. Graves, Samuel B. Why costs increase when projects accelerate. *Research-Technology Management* 32:16–18 (1989).
6. Hof, Robert D. From dinosaur to gazelle. *Business Week* Special issue: Reinventing America:65 (1992).
7. Kleinfield, N.R. How “Strykeforce” beat the clock. *The New York Times* 139(48185) Sec 3:1 (March 25, 1990).
8. Maynard, Roberta. The heat is on. *Nation’s Business* 85:14–16(1997).
9. Meyer, Marc H. and Utterback, James M. Product development time and commercial success. *IEEE Transactions on Engineering Management* 42:297–304 (1995).
10. Reinertsen, Donald G. Whodunit? The search for new product killers. *Electronic Business* 9:62–65 (1983).
11. Smith, Preston G. Your product development process demands ongoing improvement. *Research-Technology Management* 39:37–44 (1996).
12. Smith, Preston G. and Reinertsen, Donald G. *Developing Products in Half the Time*. New York: Van Nostrand Reinhold, 1991.
13. Smith, Preston G. and Reinertsen, Donald G. *Developing Products in Half the Time: New Rules, New Tools*. New York: John Wiley & Sons, 1998.
14. Von Braun. Christoph-Friedrich. The acceleration trap. *Sloan Management Review* 32:49–58 (1990).
15. Von Braun, Christoph-Friedrich. *The Innovation War*. Upper Saddle River, New Jersey: Prentice Hall PTR, 1997.