Published in *Research-Technology Management,* May-June 1992

SHORTENING THE PRODUCT DEVELOPMENT CYCLE

Here are 10 areas in which R&D managers can help to shorten the cycle time for new products and projects.

Preston G. Smith and Donald G. Reinertsen

OVERVIEW: ***As techniques like cross-functional development teams and concurrent engineering become widespread, these approaches to shortening development cycles lose their competitive edge. Decisive advantage is likely to come from the techniques competitors are not using. The authors explain that there are other untapped sources of cycle time reduction for R&D managers to exploit. These include opportunities to accelerate the "fuzzy front end," in which half of a typical development cycle vanishes before the team even starts work. The authors caution against structuring a development process around a company's largest projects because this can excessively delay smaller ones. They also question the use of phased development systems, which often cause delays in their attempts to standardize control of projects.***

The demands on product developers have never been greater. Product life cycles have shortened and continue to shrink. Product technologies, particularly in the electronics and materials areas, are changing faster than ever. As a consequence, the pressure is on to shorten product development cycles.

Frequently, the R&D manager becomes the focal point of efforts to cut development time. To deal effectively with this challenge, the technical manager must look beyond popular but limited solutions to see the breadth of the problem. Such techniques and tools as quality function deployment (QFD), simultaneous engineering, and computer-aided design (CAD) still leave many time-saving opportunities untouched (*1*).

Great reductions in cycle time are obtainable by applying various techniques blended to suit a particular company's needs. Fortunately, the R&D manager is in a strong position to initiate and foster many time-saving approaches. This article covers 10 such approaches in which such managers play a particularly important role.

1. Be Flexible About Process

There are many sound approaches to managing R&D projects. Each method has its advantages and shortcomings. The correct approach can only be selected when one has a clear vision of which advantages are critical in a particular situation. For example, consider the tradeoff between managing development time and technical risk. Most product development systems, such as phased systems (see box, page 46) are designed to monitor and control technical risk. Such systems are effective and appropriate where reducing technical risk is the paramount concern. Yet managing technical risk is not always the prime objective. Speed can be more important when trying to head off an emerging competitor, and cost can dominate our concerns in a mature market.

The most effective organizations have different development systems available and tuned to suit these distinct objectives. Without alternative processes, all projects tend to get sent through the same process, a common denominator that suits no objective well. In practice it usually errs on the side of minimizing technical risk at the expense of speed. This one-size-fits-all mentality usually creates a system tailored to the largest, most complex projects to the detriment of simpler ones. If the R&D manager is successful in minimizing product complexity, simpler systems can be used.

2. Let Economics Be Your Guide

You need a yardstick to decide which development goal to stress for a particular project, and in business the time-tested yardstick is marked in dollars. It is both simple and valuable to develop financial yardsticks for the development process. Such yardsticks tell you the relative hnancial impact of project delay versus a product cost overrun. They guide you toward choosing the most productive development goal and applying a development process that facilitates this goal.

In addition to these strategic decisions, there are countless daily tactical decisions where yardsticks help to make accurate, fast, low-level decisions on tradeoff issues. For example, is it worth spending $100 on air freight to get a sample into a customer's hands for evaluation two days sooner? How about buying an extra microscope for $2000 if it will cut a week off of the schedule? Or $50,000 for temporary tooling that will allow you to start production two months early while permanent tooling is being made? Without sound decision rules these decisions are likely to be made both incorrectly and slowly.

The financial model that provides the yardsticks is not hard to build (*2*) but it requires cross-functional effort. The finance department may have the greatest expertise in financial model building, and marketing has important

---

Preston Smith and Donald Reinertsen are coauthors of ***Developing Products in Half the Time,*** (Van Nostrand Reinhold, 1991). They are management consultants based in West Hartford, Connecticut, and Redondo Beach, California, respectively. Both have concentrated for several years on rapid product development techniques, Smith initially as staff consultant with Emhart Corporation and Reinertsen initially with McKinsey & Company. Smith received his Ph.D. in engineering from Stanford University; Reinertsen has a B.S. in electrical engineering from Cornell and an M.B.A. from Harvard.

data. But R&D should probably initiate this activity because it will obtain the greatest benefit from the model. For instance, it is not unusual to discover that the product's development expense has far less impact on its life cycle profitability than development delay. When this is the case, R&D managers who spend much of their time massaging the budget are concentrating on a low-leverage area.

3. Watch Out for Complexity

The degree of complexity in a project determines the effort needed and thus the length of the development cycle. Although this is not surprising, product development people are frequently startled to discover how quickly complexity mounts.

For example, consider the experience of a company that makes industrial process controls. They made ambitious plans to use a microprocessor in their product for the first time, figuring that this is hardly a new technology anymore. To make full use of the new capability, they tapped a new market they had been unable to serve previously, and they added new product features that had been unavailable to them before. In retrospect, they realized that complexity had multiplied on them. Microprocessors may have been familiar components to others, but this firm had to learn how to procure them, design for them, test and assemble them-everything. In addition, they had to master the new features and establish the new market. This project was slow and fraught with other difficulties.

Complexity is insidious because it multiplies quickly and its effects are indirect and often not apparent. It increases the risk-both technical and market (discussed later). It increases workloads because many more interactions among elements must be considered. It tends to draw in more people, often specialists, which complicates communication and decision making. All of this necessitates a more complicated-thus slower-management process.

The way to get new products out quickly is to minimize complexity by moving in short, simple steps, sampling customer response along the way by selling intermediate models (*2*). This incremental innovation strategy involves two roles for the R&D manager. First, as manager of technical people who generally enjoy experimenting with new technologies, the R&D manager needs to temper others' desires to put the latest technologies into new products by applying his or her accumulated wisdom. In particular, stress the use of carryover parts and standard components, such as fasteners and connectors. Emphasize inelegant but clean architectures (*2*). Rechannel the technical brainpower toward solving those problems that will provide more substantial benefit for the customer.

These days the Japanese are often held up as innovation leaders, particularly companies like Honda. In fact, these companies control their pace of innovation carefully. Even their "all new" products are often far from all new. For instance, the initial Acura automobile models which made their debut in 1986 were advertised as "new automobiles . . . designed and engineered from the driver out." However, inspection of the Acura Integra revealed that the skin was indeed new, but many functional components-the most highly engineered ones like the engine, brakes, door latches, and panel instruments-were carryovers from Honda Civics and Accords.

Complexity is insidious because it multiplies quickly and its effects are indirect and often not apparent.

4. Manage the Invention Pipeline

Complexity is minimized by moving into new areas in a planned and evolutionary way, as just covered. This does not mean that newness is avoided. Quite the contrary, newness and invention must be embraced and managed.

Invention presents a dilemma to rapid product development. On the one hand, invention is essential to innovation: continual repackaging is a dead-end strategy. On the other hand, invention is a notoriously unpredictable activity. It cannot be scheduled into a normal project, much less an accelerated one. Any attempt to schedule this wild card into a project just adds uncertainty to the schedule, and in some markets schedule uncertainty is more detrimental than a longer but certain schedule.

Resolving this dilemma falls to the R&D manager. The solution is to invent off-line in a separately scheduled program that is tightly integrated with your market and product plans. Companies like Canon, Honda and Sony are innovation leaders because they devote considerable resources to maintaining a storehouse of developing technologies basic to their businesses. Both the invention track and the product development track are market driven, and both are given resources adequate to keep projects moving swiftly. The difference is that the former is loosely scheduled while the latter is tightly scheduled. When a technology reaches the point that much of its schedule uncertainty is eliminated, it switches tracks.

Consider the two types of failures that occur when this type of system is short-circuited. Many companies try to avoid the invention track by integrating it with the product development track. Then, schedule uncertainty is high in development projects, which ultimately causes both employees and customers to have little regard for development schedules. Every project proceeds at its own pace, unable to be accelerated.

The other failure is even worse. In this case a company simply does not invent. Its development schedules are predictable, but so is its demise.

Beware of Phased Development

The concept of dividing a project into phases and funding each phase only if its satisfies certain prerequisites would appear to be a good management tool. Yet, as consultants, we see an oil-and-water relationship between phased systems and accelerated development.

Developed by NASA as the PPP (phased project planning) process, phased development systems are designed to control technical risk. But when speed is important, market risk becomes more critical: even if the product is designed according to spec, there is a significant possibility for market failure if it is introduced late. When technical risk must be balanced with market risk, a monolithic PPP-type system is no longer the clear choice. Adaptation and balance are needed in the project management system, and the balance shifts toward empowering the people and away from depending on formal control systems.

The question is really where the balance should lie. We advise some companies, usually fast-growing small ones, to formalize their review systems because with the product line and the staff growing rapidly, more formal management checks are needed to avoid technical failures. Yet in the majority of cases we see phased control systems that are overly cumbersome for a firm's needs. Sometimes a company will just adopt another's phase process, as we once observed when a loo-person instrument manufacturer adopted Hewlett-Packard's phase process, figuring that HP was also an instrument producer-a very good one indeed. But HP's process, fine-tuned for a large company with dozens of divisions, was excess baggage for this small firm.

Fundamental Limitations

Phased systems have a number of fundamental limitations that restrict the ability to shorten development cycles. They often preclude employing one of the most fundamental time-shrinking tools: overlapping activities. Often a particular stream of activities could be overlapped to advantage, but a phase review breaks the chain by requiring that all activities be finished up for review before the next phase can start.

Moreover, the very act of discovering overlapping opportunities requires a new attitude for an organization long indoctrinated in a sequential phased review process. It is that much more difficult to get people thinking creatively about overlapping dissimilar activities when their mindset is built around established phase gates.

Overlapping is enabled by employing partial, fragmentary information that evolves in a stream (2). Phased systems fight against partial information, providing credit and passage to the next phase only when the information package is complete. Here again, the time-saving opportunities must be discovered in a particular circumstance by thinking creatively about inching forward with the information at hand. A tidy phase framework discourages these discoveries.

Exploiting system architecture opportunities is another means of compressing development cycles (*2*). By dividing a product into subsystem modules with relatively clean interfaces and ample performance margins, these modules can be developed concurrently by different teams. For example, a transceiver might be divided into a power supply, a transmitter, a receiver, and an audio amplifier. There is no reason to believe that these four modules will have the same timing. One module might require more conceptual design or technology exploration, while another might need a great deal of prototype testing. Putting all modules in a lockstep phase review process stretches the overall cycle.

Moreover, a phased system encourages queues. Queue reduction is a huge and inexpensive opportunity to shorten cycles simply because most development projects spend the majority of their time sitting in queue somewhere. With phases, queues build up in preparation for a review. (Remember that only<a complete package is acceptable for a review, so some items wait while the package is completed.) Then, when the review is complete, a flood is released into the next stage of the system, swamping it.

For example, upon final approval of the design, purchasing may be faced with simultaneously ordering a million dollars of capital equipment from a dozen suppliers. Or in one case we observed, the chief engineer signed and released into manufacturing nearly 500 drawings in a single day when his project passed a milestone. You could actually observe the glut passing through the manufacturing transition process.

A phased process also causes problems when, as is often the case in practice, "the product" is really a line of products in different sizes, materials or colors. Then, forcing all variants into synchronization for convenience in review creates both pre-review delays and post-review gluts unnecessarily.

Responsibility Belongs To People

In short, phased approaches are attempts to build judgment into the process rather than into the people. It is reactive, it is slow, and it removes the responsibility from the people doing the work, where the responsibility belongs.

Shifting away from a phased process is difficult. There is a great deal of management comfort involved in taking a thorough, formal look at a project periodically and making an explicit decision whether or not to proceed. Unfortunately, the cost can be high when time is at stake. A balance must be struck between comfort and speed, and all too often, comfort wins out even when speed is the key competitive factor.

Some rapid product development specialists suggest that the phased approach should remain the foundation but that the phases should be compressed and "dead time" between phases should be eliminated (*5*). Our observations of how development projects actually proceed in industry suggests that the greatest opportunities for improvement lie in eliminating the delays associated with synchronization and queueing. This requires a fundamental departure from the phased approach, not fine-tuning it. - **D.R. and P.S.**

5. Avoid the "Thinking Stage" Trap

Other departments are often quick to blame R&D for slow product development, but the fact is that half of the typical product development cycle has vanished before development is even authorized (*2*). What we call the fuzzy front end is frequently one of the largest and cheapest opportunities to shorten the development cycle.

The front end of a development project starts when the need for a new product is first apparent, whether the company acts on it or not. Product need could be mandated by the enactment of a new government regulation, the emergence of a new technology, or certainly, the appearance of a competing product. The front end terminates when the firm commits significant human resources to development of the product.

We are not suggesting that the front end is unimportant; it is more like the heavens-mostly empty. Some crucial decisions are made during this period regarding the size of the market opportunity, the target customer, alignment with corporate strategy, and availability of key technologies and resources. In fact, research on the market success of new products suggests that products fail because companies don't do enough of this "homework" (*3*). Nonetheless, front-end time is still mostly a vacuum, largely because managers who haven't calculated the dollar value of development delay believe that time is free until people are assigned to the project.

As an R&D manager, your role in this phase is to be hard-nosed about using your people on product concepts. Resist the attempts of marketing or general management to have one of your people "look at" an idea in their "spare time." Remind them that delay erodes product profitability, and offer to assign one of your people immediately at full- or half-time to reach a certain decision by a definite date. If you aren't this serious about using your resources, then the company isn't serious about the product concept.

6. Staff Teams Adequately

Our experience, and that of many others, suggests that product development proceeds most quickly and effectively with a team of six to ten full-time members. Although some products, like automobiles, computers or aircraft, require more effort than this, far more common is the development project that seems too small to justify this level of commitment. It receives perhaps a full-time person, a couple of part-timers, and a flock of bit-part participants. Given the heavy load of projects underway at a typical company, this appears to be the best that can be done. No single project has enough importance to command adequate resources.

The solution to this situation is simple in principle. If each project requires a certain number of person-years of effort, consider doubling the staff on half of the projects and complete them in half the time. Then do the other half of the projects similarly. Fewer projects will be underway at any point in time, but the same number will be completed each year.

Half of the typical product development cycle has vanished before development is even authorized.

Although the annual output is the same, the shorter, more intense project option has several benefits. The projects started first get to market sooner, giving them a competitive advantage and a longer sales life. The ones started later are completed no later than before, but they enjoy the advantages of a late start, such as better market information and more recent technologies. Both the early starts and the late starts reap the advantages of a short cycle: fewer opportunities for the market or the project objectives to shift, which means less redesign.

The shorter, more intense option is a vatid model if project pacing is primarily dependent on labor availability, that is, project tasks typically sit waiting for people to work on them. In our experience, this is a frequent occurrence. Occasionally, project pacing depends primarily on outside events, like tooling lead time or prototype testing time, in which case it may not be possible to save appreciable time through heavier staffing.

A common objection here is that a more intense effort suggests large teams and thus a greater communication burden, which negates part of the anticipated benefit. However, we can obtain the extra effort without extra people by staffing teams with full-timers rather than part-timers.

Although it is possible to overstaff a team, our experience suggests that American development teams suffer much more from fragmented understaffing than from overstaffing.

7. Staff with Generalists

Often teams are fragmented, having many part-time members, because people are viewed too narrowly, and they in turn often mold themselves as narrow specialists. Some people indeed must be highly trained in a specific technical area in order to advance the state of the art, but the need for such skills is limited in most product development, which instead stresses application and integration. Having a narrow person on a development team causes the R&D manager three problems:

- First, it is difficult to keep such people fully occupied on the project. They tend to split their time commitment among one or more other projects. They drift in and out of a project as their particular skills are needed or as they have the time. They may shift to another project when the going gets tough on a

particular project. Consequently, the team leader is consumed by simply keeping the team together and communicating with the part-timers, not on the primary task of developing the product.

- The second difficulty with specialization on the team is that good products require balance to provide value to the customer. This balance is achieved most quickly and with the least communication burden if everyone involved has a solid appreciation of the customer, the economics, the various technologies involved, and the manufacturing methods.

- Third, a high degree of specialization inhibits the manager's ability to redeploy people within a development team to match the workload, which leads to queues and delay.

There are a couple of implications for the R&D manager. First, staff teams with generalists or those willing to become generalists. This will give the team a more comprehensive view, which will allow them to move quickly and precisely. It will also strengthen the team if team members can shift to various secondary team jobs rather than dropping off of the team for a while.

Second, encourage and develop generalists. We recently saw a manufacturing engineer take on the company's cost accounting system when the manufacturing costs for his product weren't coming out to his liking. He didn't overhaul the corporate system, but he did negotiate a more equitable way of costing his product, a more creative one than a cost accountant was likely to have proposed. He learned a lot about cost accounting in the process, and he is now more valuable to his organization.

Deliberately expose people to new areas, either by transferring them to new departments or through outside training. Send your engineer to an accounting course or your draftsperson to a production machine programming course; encourage a marketing person to enroll in a microwave fundamentals course. Just by staying on a development team from start to finish, people will broaden, but this process can be accelerated through deliberate training.

8. Let the Team Manage the Team

Product development is just a succession of problems to be solved, so development speed depends on the speed of the problem-solving process, which in turn depends on how tightly problem-solving loops are connected. Every time the team has to go outside of itself to obtain a decision, additional delay is incurred. The farther it has to go, geographically or organizationally, the greater the delay is likely to be.

A Boston-area computer peripherals firm attacked the problem-solving-loop issue directly. The vice president observed that their development team was wasting time because designers weren't getting good enough guidance at their weekly meetings. They would design what they thought was desired, only to find out a week later that they were off track and most of the week had been wasted. So the vice president organized short, daily team meetings at the ten o'clock coffee break. Not only was there less waste of design talent, but everybody moved faster and more surefootedly because progress was now measured on a daily basis.

Product development proceeds most quickly and effectively with a team of 6-10 full-time members.

Just as important as the fivefold shortening of the loop is the vice president's role in this process. He recognized the problem, got the group to meet daily, and even attended many of the meetings. But he didn't run the meetings or participate in their content. He only made sure the group got together daily and left each meeting with a clear idea of what they would be doing next. The team ran the team, and if the company had had a team leader, the vice president's involvement would have been unnecessary.

Getting the team to run itself involves a couple of difficult organizational challenges. In typical organizations, managers of R&D and other functions typically control pieces of development projects, which are completed as these managers coordinate their efforts-often slowly. These roles must shift as the team assumes control of the project. The functional managers now act as advisors and coaches, assisting their colleagues on the team but not using the colleagues as conduits to carry information back to the functional manager for a decision. If this occurs, the organization has just installed a group of puppets as an additional level in the decision-making process. The R&D manager will still have plenty to manage, but his or her role changes with respect to a fast development team.

Frequently the team is also uncomfortable with its new role of making final decisions. If it tries to toss the decision back to management, management must simply toss it back to the team. Before long the team will rely on management as a source of decision-making information, not as a source of decisions.

9. Manage Both Technical and Market Risk

Risk management has always been a large part of the technical manager's job. As cycle length shrinks, this job becomes even more essential. Faster projects employ a greater amount of task overlapping, which creates loose ends and, in turn, more opportunities for key steps to be omitted accidentally (*2*). If the team is managing itself, as just suggested, there is less formal opportunity for the management hierarchy to apply its considerable experience to averting past mistakes. Finally, there is less time available in a compressed schedule to recover from problems.

Fortunately, there is a great deal the R&D manager can do to manage risk in an accelerated project. Sensitivity to risk comes in part from years of experience, which managers are more likely to possess than are the members of the team. Through frequent informal interactions with the team, management can see potential pitfalls and inject insight to cope with them, all without infringing on the team's charter to make its own decisions.

One area where the manager's experience is most valuable is in balancing testing and analysis. Many technical people are prone to analyze an issue profusely before building something and testing it. Just making a model seems like an unprofessional expedient, but expedients are often just what we are looking for as we try to shrink tasks. Others, who may lack the analytical skills or discipline, do the opposite. They build and test repeatedly before thinking much about what the underlying issues may be, so they waste time in resolving risk, too.

The trick is in knowing when to test and when analysis would get the answer faster, or better yet, how test and analysis can be blended to get the best of both. The R&D manager's accumulated wisdom can be invaluable in raising and helping to resolve these issues. The manager also must make sure that analytical and testing resources, such as an open lab, are easily available to the team for this hands-on work.

Risk is of two types: technical risk, which is the inability to satisfy the product specification, and market risk, which is the inability to sell the product assuming it meets specifications. We tend to concentrate on technical risk, ignoring market risk, because we have better techniques for resolving technical risk, it is easier to identify and measure, and its symptoms usually appear sooner.

The R&D manager's job here is to teach the rest of the organization that market risk is just as real as technical risk and that the same general risk management techniques apply to it, although the two should be managed independently (*2*).

10. Develop a Reserve

We have saved the toughest topic until last. As suggested in 6, above, development projects are slow largely because they spend most of their lives waiting to be worked on. Projects are abundant but resources are tight. One reason for this predicament is that we use the popular development funnel concept where it doesn't apply.

For some products, often chemical products, the concept of a development funnel does make sense. The failure rate in the initial feasibility stages of a project is high, and the cost of these stages is low. So we start lots of projects at the top of the funnel, and a few winners flow from the bottom through a natural selection process.

Ironically, the development funnel doesn't work well for many products because the failure rate isn't high enough. Such projects are more likely to succumb to market causes either before or after development than to fail on technical grounds during development. Nevertheless, companies load the funnel with plenty of new-product ideas, and marketing is in fact encouraged to overstock the funnel (**2**). Because few projects actually fail, projects languish in the funnel awaiting resources. R&D managers must discourage application of the development funnel mentality where it does not apply. Applying it under low failure-rate circumstances generates a glut and demoralizes technical people whose perfectly acceptable projects get shelved in midstream for lack of resources.

American development teams suffer much more from fragmented understaffing than from overstaffing.

However, eliminating just the glut is not going far enough. There actually has to be some slack because unplanned new product ideas will arise unexpectedly. The time-competitive firm needs some reserve development capacity to respond to these customer needs quickly, just as they retain reserve manufacturing capacity to fill unanticipated production orders responsively.

This presents a difficult challenge for the R&D manager. At planning time, don't accept a full load and then a bit more to cover fallout. Instead, leave some unused capacity for the really new projects.

This completes our tour of 10 areas where the R&D manager can shorten development time dramatically. You obtain the greatest benefit by making all of these improvements, because they all reinforce one another. But this is a long-term goal because no company we know of does all of these things well yet. So get started with some of them, perhaps by using a pilot rapid development project to initiate several of the changes immediately (*2*). Finally, get the non-R&D parts of the company involved compressing development time too. Even those apparently removed from the process, like corporate planners, have essential parts to play (*4*). ◭

References

1. Reinertsen, Donald (1991); "Outrunning the Pack in Faster Product Development," ***Electronic Design, 39,*** 1, pp. 111-118.
2. Smith, Preston, G., and Donald G. Reinertsen (1991); ***Developing Products in Half the Time (New*** York: Van Nostrand Reinhold).
3. Cooper, Robert G. (1988); ***Winning at New Products*** (Reading, Mass: Addison-Wesley).
4. Reinertsen, Donald G., and Preston G. Smith (1991); "The Strategist's Role in Shortening Product Development Cycles," ***The Journal of Business Strategy,*** July/Aug. pp. 18-22.
5. Rosenau, Jr., Milton D. (1990); ***Faster New Product Development (New*** York: AMACOM).