# A Total Systems Life Cycle View on Reducing Cycle Time

## "Three Lenses Provide the Right Customer Focus"

LT. COL. BRIAN BRODFUEHRER, U.S. AIR FORCE

As far back as the 1986 Packard Commission, reducing cycle time was recognized as crucial to any genuine reform of DoD's acquisition system. According to the Commission's report, *A Formula for Action*, "An unreasonably long acquisition cycle – 10 to 15 years for our major weapon systems ... is a central problem from which most other acquisition problems stem:

- It leads to unnecessarily high cost of development...
- It leads to obsolete technology in our fielded equipment...
- It aggravates the very gold plating that is one of its causes...."

This article looks at cycle time reductions from a *total systems life cycle* perspective. The total systems life cycle perspective ensures that short-term reductions in the development cycle are not lost later through delays in maintaining and modifying the system. Such short-term reductions and savings may lead to unsatisfied customers and higher long-term operations and support costs.

Also in this article, I describe industry best practices, providing a systems view of cycle time reduction, including a list of tools to apply and a list of factors that influence applications of the tools. From my perspective as an instructor at the Defense Systems Management College, this article continues my efforts to emphasize and support development of creative problem-solving skills for application to program management scenarios requiring quick reaction and astute change management.

### Framework for a Total Systems View

"Reducing Cycle Time" means providing a capability to a customer in less time. In the commercial arena, reducing cycle time might mean getting a new product to market in less time than the previous product version, important because of the need to stay ahead of the competition. The time to get that product to market, the macro-cycle, is made up of micro-cycles all contributing in some way to that top-level time metric. These micro-cycles, or sub-processes, consist of different activities that, together, make up the product development process. These include such functions as requirements definition; the analysis and decomposition of the requirements into designs and drawings; and the production and testing of systems (both hardware and software) for delivery to the customer. Looking at the DoD system in an analogous way, the macro-cycle is the time it takes the acquisition community to deliver supportable products to the customers, the requirements community (users), or the warfighters.

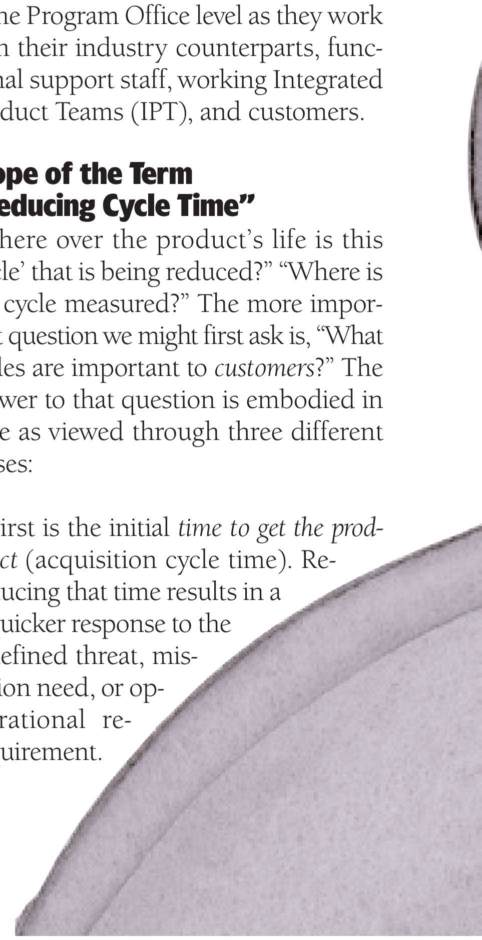
Three interlinked systems – the acquisition process; the requirements process; and the Planning, Programming, and Budgeting System process – define this macro-cycle. The most effective treatment of cycle time reduction would address changes at the macro level of each of these three different systems. However, making changes at a level encompassing these three systems is generally out of the scope of influence of most program managers; thus, the focus of this article is on achieving cycle time reductions *within* the constraints of the three macro-systems, not from trying to *change* those systems.

The process program managers can best influence is the acquisition process of the specific systems they manage. Focusing on cycle time reductions at that level can contribute to an overall reduction in the time it takes to deliver a capability to the warfighter. Ideally, these reductions will be achieved by managers at the Program Office level as they work with their industry counterparts, functional support staff, working Integrated Product Teams (IPT), and customers.

### Scope of the Term "Reducing Cycle Time"

"Where over the product's life is this 'cycle' that is being reduced?" "Where is the cycle measured?" The more important question we might first ask is, "What cycles are important to *customers*?" The answer to that question is embodied in time as viewed through three different lenses:

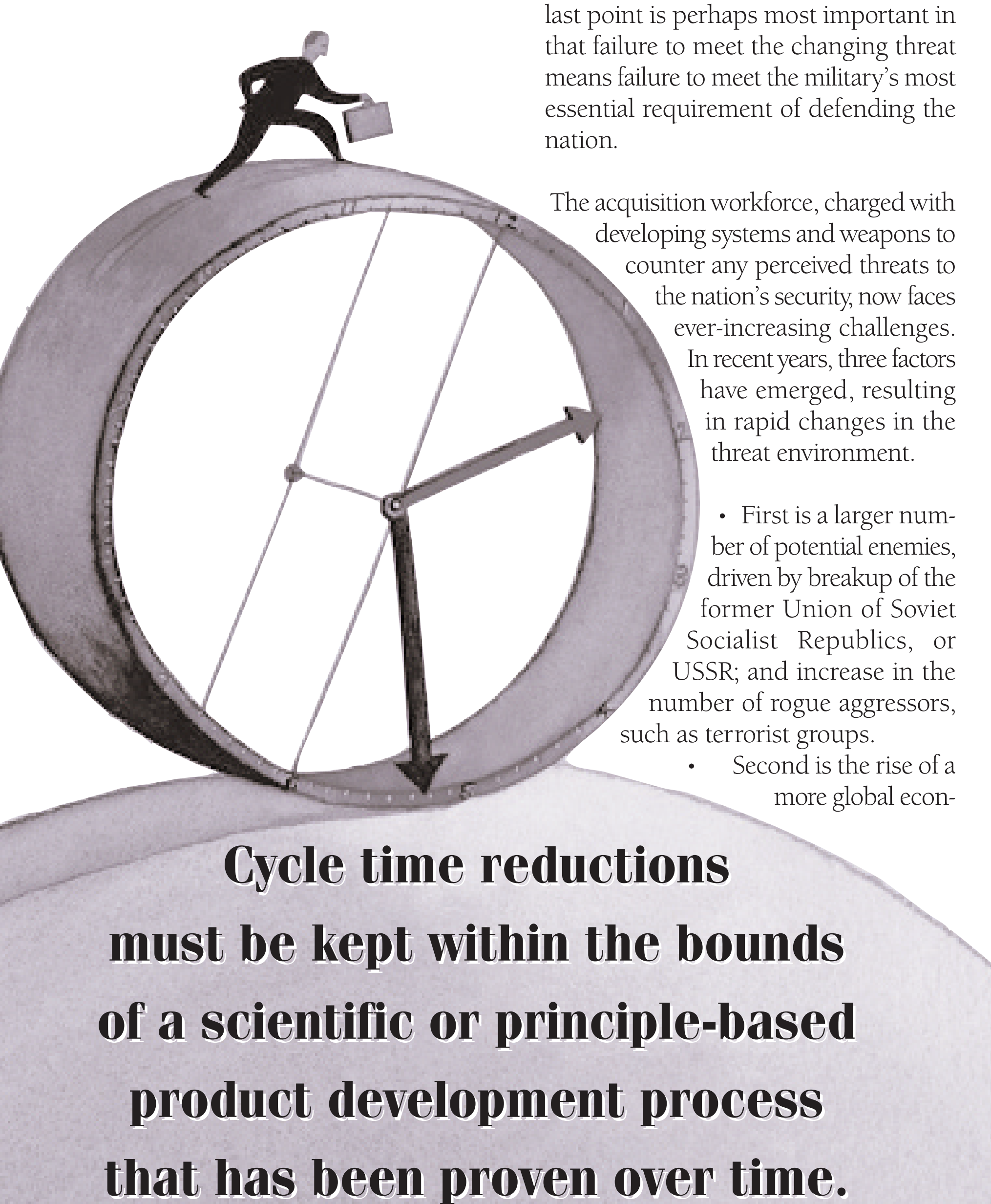
- First is the initial *time to get the product* (acquisition cycle time). Reducing that time results in a quicker response to the defined threat, mission need, or operational requirement.



**Brodfuehrer** *is a professor of Systems Engineering Management, Systems Engineering (SE) Department, Faculty Division, DSMC. He is a graduate of APMC 88-2, DSMC.* *The author wishes to acknowledge the SE Department and the Problem Set (PS) 805 Team for their support and insights during the preparation of this article.*

- Second is the *time it takes to support* (maintain and repair) the system (logistics cycle time).[1] Reducing that cycle time improves the availability of the system to support mission requirements quickly and consistently.
- Third is the *time it takes to improve or upgrade the system* to respond to new threats or requirements, to fix system shortfalls, or to improve system reliability (evolutionary cycle time).

Therefore, when an acquisition strategy is developed to "reduce cycle time," it should, at a minimum, address issues that arise when looking at the problem from at least these three views. Understandably, delivering a product to warfighters quickly but leaving them with a system that 1) does not meet performance expectations; 2) cannot be easily maintained; or 3) cannot be improved when the threat changes, has not effectively reduced cycle time from the long-term perspective.



**Cycle time reductions must be kept within the bounds of a scientific or principle-based product development process that has been proven over time.**

## Motivation to Reduce Cycle Time

In a competitive commercial sector, taking too long to get a product to market can drive a company out of business, as will taking too long to support or improve the product. In DoD, prolonging the time required to meet users' requirements lowers customer satisfaction, costs more money, and is not responsive to the changing threat environment. The last point is perhaps most important in that failure to meet the changing threat means failure to meet the military's most essential requirement of defending the nation.

The acquisition workforce, charged with developing systems and weapons to counter any perceived threats to the nation's security, now faces ever-increasing challenges. In recent years, three factors have emerged, resulting in rapid changes in the threat environment.

- First is a larger number of potential enemies, driven by breakup of the former Union of Soviet Socialist Republics, or USSR; and increase in the number of rogue aggressors, such as terrorist groups.
- Second is the rise of a more global economy resulting in less restricted sharing of technology.
- Third is the rapid growth of technology, particularly in the computer and communications industry sectors.

These three factors combine synergistically to make future threats increasingly elusive and powerful. Altogether or perhaps even separately, these threats will continue to churn a rapidly changing environment – one requiring quick, effective responses to maintain a secure national defense.

## Technology Insertion and Reduced Cycle Times

Effective application of new technology is one force multiplier that the United States uses to its advantage in countering threats, dominating the battlespace, and maintaining information superiority. New technology, when properly inserted into a program, can improve both performance and supportability.

One characteristic of new technology (especially in the electronics, information systems, and communications areas) is that it changes so rapidly. Keeping up with these changes is vital, both from the perspective of knowing what our global competitors are doing with the technology, and of knowing how to best apply advanced technology to serve our nation's interests.

When technology platforms change significantly every 18 months, but we [DoD] cannot deliver new capability any faster than 10 years out, we fail to leverage the improved capabilities of advanced technologies. Reducing cycle time enables effective use of new technology for the warfighter.

To effectively leverage new technology, program managers should first understand the DoD process for developing new technology and transitioning it into the warfighting arsenal. Understanding the process is a critical tool to reducing cycle time. This process is covered in the Defense Systems Management College Advanced Program Management Course (APMC) and Advanced Systems Planning, Research, Development and Engineering Course (ASPRDEC) curricula, among others.[2]

## Cautions on Reducing Cycle Time

When reducing cycle time, be careful – cut the fat, not muscle or bone. Neither the acquisition community nor the warfighters are satisfied by serving up platitudes like doing "more with less" without identifying a realistic, prioritized approach as a guide.

Recent failures in the satellite and launch community and in the National Aeronautics and Space Administration Mars program have raised questions about the way the "better, faster, cheaper" approach was implemented in those arenas.[3] Shortcuts that omit important technical management activities may be false economies. Likewise, cycle time reduction must be done in a way that is intuitively supported as being common sense at all levels of the organization. Leadership's most effective improvements come from making strategic decisions on *what* to work on and *what not* to work on. So it is with cycle time. Workers' improvements come naturally from finding better ways of doing the job assigned.

Reductions must be kept within the bounds of a scientific or principle-based product development process that has been proven over time.[4] Use prudence in tailoring this process for a particular program; do not compromise the disciplines that define the process.

## Metrics

Four metrics are available for program managers to measure the success of cycle time reduction strategies:

- First is *time*. Does the strategy effectively reduce the time in the three dimensions described earlier – acquisition, logistics, and evolutionary?
- Second is *cost*. Does the strategy consider the life cycle or total ownership cost of the product? This includes development, acquisition, operations and support, and disposal costs.
- Third is *customer satisfaction*. Does the system meet requirements? Would the customer come back or recommend your service to another customer?
- Fourth is *resource management* (money or manpower). Are your people taken care of; would they work with you again on another project? Is the project well managed? Would your boss give you another project? This last metric measures the resilience of the acquisition management system, and its ability to continue to support the warfighter and the taxpayer at a particular operations tempo.

## Cycle Time Reduction Tools

Several cycle reduction tools are available for program managers. For purposes of this article, I grouped these tools into three categories originating in a study on defining next-generation products.[5] In this study, two Silicon Valley consultants (Tabrizi, Walleigh) examined 28 next-generation product-development projects and identified best practices that led to success. These practices were then placed in three categories: product strategy, project organization, and execution in the definition phase.

The following discussion on tools applicable for reducing cycle time with a total systems approach, uses that study as a framework. The best practices from the study have been converted to tools here and were also adjusted so that they conform to DoD-applied terminology. Other best practices were added to the framework, where appropriate.

### Product Strategy Tools

#### Work Closely with the User

Maintain a continual dialogue with the customer or user, including the entire customer base – the maintainers and trainers, in addition to the operators. Share information on technical trends, updates on progress, and possible applications of new products. Understand how the user will use and support the system. Talk to users about the current systems to understand their shortfalls. Excellent communication with the customers early in the product strategy phase will result in less scrap and rework later in the process, less time and cost spent in Initial Operational Test and Evaluation, and higher customer satisfaction.

#### Use Cost As an Independent Variable (CAIV)

This tool is a framework to bring cost constraints into any dialogue with the user. It requires setting aggressive cost objectives early in the process of defining a new product or of changing an existing product. The acquisition and requirements community must work closely to identify options and make trades between performance and cost. Effective use of Cost As an Independent Variable (CAIV) can reduce cycle time by proper setting of expectations early in the process and requiring less scrap and rework at the end. Design to Cost is a sub-tool that can be used to implement top-level CAIV objectives by allocating those constraints to lower levels of the program.[6]

#### Develop an Integrated Road Map

The road map contains at least two levels:

- How, over time, does this system fit with other systems with which it is to operate?
- How, over time, will this system evolve and integrate its various sub-components?

The road map should be a living document, updated at regular intervals or when major externally generated changes occur. This tool saves time by smart, forward-thinking change management. Changes in the form of new requirements, diminishing parts sources, software upgrades, supportability upgrades, and changing technology will impact the program. Planning to bundle those changes into discrete blocks at regular intervals will reduce the number of blocks required, thus reducing the testing required and the management of different configurations in the field. This reduces both the acquisition and supportability time.[7]

#### Use an Evolutionary Approach

Evolutionary development or acquisition consists of working closely with the users to deliver increments of capability in complete, discrete, and expanding cycles. The first cycle addresses the mission deficiency as currently known today or some portion of that deficiency. A system is defined, built, tested, and fielded in a rapid but controlled manner. The control is applied by following a systems engineering process. The evolution continues to the next cycle as changes such as new threats or technology improvements or sustainability needs arise. This cycle repeats the same process, accounting now for the new information and requirements. Additional information about this evolutionary approach appears at the end of this article.[8,9]

### Develop Complete Option Strategies for the User

Options developed to reduce cycle time should consider the three time dimensions discussed earlier (acquisition, logistics, evolutionary). Systems delivered quickly but with high maintainability and upgrade overhead are of limited value in today's environment of rapidly changing technology. Initially, they may appear to have reduced cycle time, only to revert to much longer cycle times downstream. Options developed for the user should identify life cycle cost and ease with which a system can be upgraded. Downstream producibility, supportability, and "upgradeability" constraints should be included as part of the requirements definition and design processes—again, this points to using a systems engineering process. The user, or customer, should be advised of these three long-term aspects of each option offered, in addition to the immediate cost and schedule.

Building effective option strategies requires training of both engineers and marketing personnel in the up-front design factors that impact these downstream product characteristics. Company design manuals should contain techniques that show the impact of today's decisions on tomorrow's customer operational cost and workload. Customers then become informed consumers, and can adjust the long-term road map if today's cost and schedule constraints require delivery of a less supportable or producible system.

### Use Open Systems Architectures (OSA)

The Joint Technical Architecture (JTA) sets the standards for DoD communications — standards absolutely crucial to DoD's mission of protecting the nation. Such standards must promote interoperability among the Services, a required characteristic for joint and allied operations. The JTA is continually evolving and is currently at Version 3.[10]

**The process program managers can best influence is the acquisition process of the specific systems they manage. Focusing on cycle time reductions at that level can contribute to an overall reduction in the time it takes to deliver a capability to the warfighter.**

Key to DoD's interoperability initiative is a design (and business) tool that recognizes and accommodates change called Open Systems Architecture (OSA). An OSA places the design focus on commonly used and widely supported interface standards. Think of OSA in terms of the axle-wheel-tire interfaces employed on commercial cars. By adhering to common standards at the interfaces, the consumer can buy tires from a multitude of suppliers, rather than being forced to buy from a single source, as might be the case if the interface characteristics were unique."[11] This approach can also be seen in electrical wall sockets, VHS video recorders, and personal computer clones.

Moreover, using OSA reduces cycle time and effectively leverages technology from the commercial marketplace. Commonly used interface standards permit several suppliers to provide products instead of tying the customer to only one.

In some industrial sectors (such as information technology), the commercial marketplace is driving technology improvements at a much greater rate than in the defense-related industrial base. OSA enables the use of this technology. It also extends the usable life span of major weapons systems (such as the B-52 or C-130) by facilitating the upgrade of avionics systems. It takes much less time to modify an F-18 or F-16 than it does to build a new fighter.

### Use Product Platforms for Software Development

This is a tool that ties in well with OSA and also accommodates changes. Meyer and Seliger define a product platform as a set of subsystems and interfaces that forms a common structure from which a stream of derivative products can be efficiently developed and produced.[12] These are also called reference architectures. The efficiency is measurable in terms of cost and time required to generate products from underlying platforms. A set of computing infrastructure and application building blocks form a platform from which springs any number of different applications that can be applied to different markets or customers. The time to produce a capability is thus reduced because of the existence of the common platform, enabling the development effort to focus only on the different application software.

### Fill in the Gaps Between Evolutionary Capability Increments

Between one large increment in capability and the next will be performance gaps. These gaps represent valid user needs that may go temporarily unfulfilled. There might also be gaps in supportability or training that need to be filled in with finer increments leading to the next evolutionary block. Companies successful in the commercial sector at developing new products fill gaps between platform releases with "derivative products."[13] This concept could be applied to the defense sector by providing interim releases of minor support products to aid in the existing performance shortfall until the new block evolves.

## Project Organization Tools

### Form the Team Based On the Difference Factor or "Newness" of the Product

Capturing new markets or developing systems that represent a significant leap in performance could well require a new and separate group to develop. A new team could also be required if the culture required to produce or operate the system (due to new technology) was significantly changed. If established cultures and processes will not support the new product, then establish a new team in an environment that will support growth of the new culture. An existing team could well handle smaller changes in performance.

### Phase the Team's Staffing Mix

Too many engineers early in the project could result in excessive and frustrating "wheel spinning" while the requirements are in flux. Too few later in the project result in work overload. One successful company adds new people to the original small team after it completes the initial specifications. Typically, successful companies use just a few senior experts during the initial phase of the program. Senior marketing experts with a good grasp of both technology and the market work with senior technical experts who have a good grasp of risks, technology obstacles, cost and time constraints. Their experience and influence enable ideas developed early on to carry through to the product development phase, thus reducing time.

## Execution During Definition Tools

### Track Progress and Sustain Urgency

The study found that the root causes of delays were managerial in nature: lack of processes to monitor time and paying insufficient attention to the routine details of the product definition process. A disciplined systems engineering process integrated with an earned value management system provides an effective way to track progress. Other tools leading to success follow.

### Use A Product-Priority Document

The customer's product-requirements document (the Operational Requirements Document in DoD acquisition terminology) is prioritized into categories such as "must have," "should have," and "nice to have." Thresholds and goals would be two applicable categories. This prioritization supports trade-off discussions with the user that might go as follows: "If we add this feature, our cost will grow by x dollars and our development schedule will be slowed by y months. Are you willing to pay more and wait longer?"

Two tools discussed earlier in this article – CAIV and building complete option strategies – can aid these discussions with the users.

### Develop Early Prototypes

Successful companies move quickly to prototype key subsystems and then the entire system. These more realistic representations of the system energized the development team and enabled fruitful, more focused discussions, resulting in quicker decisions. Typically, these companies involved customers with the evaluation of the prototypes and used their comments to converge on the final product design. "The customer dialogue does not delay product development. Rather, it provides a continuous stream of market information that helps shape derivatives and revisions."[14]

### Demonstrations

The best practices of early prototypes are analogous to the DoD practice of using demonstrations. DoD prefers these methods of assessing and reducing concept risk and assessing military utility of alternative technologies.[15]

### Use Development Partnerships

Successful companies partnered with suppliers or with other companies that brought technical or financial strengths to the partnership that they lacked. Together, they handled disagreements at the working level, not "by contract amendments or litigation."[16] Sharing people and technology allowed differences, whether in specifications or culture, to narrow. Likewise, DoD program offices can partner with other DoD organizations, e.g., laboratories, to take advantage of specific strengths that the program office may lack.

### Develop a Two-Track Strategy for Primary Systems and Derivative Products

Tabrizi found that major system changes, whether new starts or major modifications, or large evolutionary steps tend to have: high uncertainty, specifications that need to evolve over time before they are finalized, initial staffing requirements met by a few key people, and fewer milestones required up front for effective tracking. Alternatively, derivative products (sometimes referred to as "precedented systems" or gap filling changes) tend to have low uncertainty, specifications that are defined quickly, higher staffing requirements, and more detailed, closer-spaced milestones required for effective tracking.

The differences in these two types of developments should be reflected in the strategies placed in the product development road map. Tailoring the development strategy to the size of the development can reduce time required to develop small increments or product derivatives and will appropriately adjust the time required for larger leaps of change.

## Factors Influencing Tool Selection

The tools/best practices listed in this article are provided as options to work the problem of reducing the time it takes to deliver a capability to the warfighter. Applying the right tool to fit the circumstance is also important. A wrench can be used as a hammer and a screwdriver as an awl, but it will likely take longer and not be as effective.

This section discusses factors to consider when selecting tools to reduce cycle time. All these factors will impact the selection of acquisition strategy, the implementation of strategy through the selection of the tools discussed earlier, and the tailoring of the acquisition process to the needs of the specific program.

### Phase of the Acquisition Program

Is this program a new start, one that DoD has already fielded, or one that is being modified? Many of the tools discussed earlier apply to all three situations. However, OSA can be applied more readily to a new start than to a program that has already been designed. Certainly, use of OSA on a fielded system will require more thought on the part of developers and users. For instance:

- How long will the system continue to be fielded?
- What is the cost of a new architecture?
- Can the architecture be upgraded incrementally?

Another key aspect of working with an existing program is that any changes must be carefully worked into the stream of the ongoing program.

- If the program is in production, how will the change be incorporated into the factory floor?
- If it is already fielded, how will changes be made?
- Will they be made in a depot or by maintainers in the field?
- How long can the systems reasonably be out of service?

### Impacts of Change

Two aspects of change are influential when deciding which tools to select. The first is the rate of change of the environment. Faster changing environments require shorter response cycles; more time is available when the threat is not changing as rapidly The second is the degree of change from cycle to cycle. Risks are higher for larger changes, and more time may be required to complete a successful program.

### Maturity of the Technology

Assessing maturity of the technology being inserted is an important part of determining risk associated with the change. This risk assessment will impact the selection of a reasonable time for delivery of the change to the user. Inserting immature technology will increase cost and schedule risks of the development and production program as well as drive up operations and support costs.

## Learning and Applying

This article looked at cycle time reductions from a total systems perspective. Cycle time reductions not well thought out early in the definition phase are likely to be swallowed up by large delays and unsatisfied customers later on. Rather than simply disseminate policy statements on cycle time reduction, my intent was to inform and challenge students (and *PM* readers) to learn and apply specifics based on industry best practices in their own programs.

In summary, a thorough understanding of a systems life cycle view of cycle time reduction, including all three lenses of cycle time reduction (acquisition, logistics, and evolutionary) is critical to reducing cycle time. Armed with tools that are based on industry best practices, the acquisition workforce at large, I believe, can better develop and apply the strategies discussed in this article to effectively reduce cycle time.

**Editor's Note:** The author welcomes questions or comments on this article. Contact him at **brodfuehrer_brian@dsmc.dsm.mil.**

## References

1. "Logistics Cycle Time" includes all time consumed by logistics actions such as: Mean Time Between Failure (MTBF), Mean Maintenance Time (MMT), Mean Time Between Maintenance (MTBM) (includes all corrective and preventive maintenance tasks), Logistics Delay Time (LDT), Administrative Delay Time (ADT), Mean Time To Repair (MTTR), and Mean Logistics Down Time (MLDT).
2. Zittel, Randy, "Overview of the DoD Science and Technology (S&T) Program," Advanced Systems Planning, Research, Development, and Engineering Course (ASPRDEC) Teaching Note, DSMC, February 1998.
3. Sawyer, Kathy, "Please Phone Home," *Washington Post*, Dec. 5, 1999.
4. Electronic Industries Association (EIA) Standard IS-632, Processes for Engineering a System (ANSI/EIA-632-98), Jan. 18, 1999; or Institute of Electrical and Electronics Engineers (IEEE) 1220, Standard for Application and Management of the Systems Engineering Process, 1998.
5. Tabrizi and Walleigh, "Defining Next-Generation Products: An Inside Look," *Harvard Business Review,* November-December 1997.
6. DoD 5000.2-R, *Mandatory Procedures for Major Defense Acquisition Programs (MDAP) and Major Automated Information System (MAIS),* Section 3.3.4, "Cost As an Independent Variable," May 11, 1999.
7. Aeronautical Systems Center/Avionics Engineering (ASC/ENA) Briefing on Reducing Total Ownership Cost (TOC), Bob Gibler et al, "Avionics Systems Architectures and Cost," Synchronizing Change Section, circa Summer 1999.
8. Defense Acquisition Deskbook, *Evolutionary Acquisition*, Dave Anderson, September 1998.
9. Defense Acquisition Deskbook, *Using Evolutionary Acquisition for the Procurement of Complex Systems,* Henderson and Gabb, Australian Defence Organization Report DSTO-TR-0481, March 1997.
10. Joint Technical Architecture, Version 3, Nov. 15, 1999 (**http://www-jta.itsi.disa.mil**).
11. Hanratty, Lightsey, Larson, "Open Systems and the Systems Engineering Process," *Acquisition Review Quarterly,* Winter 1999.
12. Meyer, Seliger, Sloan, "Product Platforms in Software Development," *Management Review*, Fall 1998, p. 61.
13. Tabrizi, p. 119.
14. Tabrizi, p. 123.
15. Office of the Secretary of Defense, Under Secretary of Defense (Acquisition & Technology) Policy Letter, "Implementing Cycle Time Reductions," July 9, 1999.
16. Tabrizi, p. 123.