

Modern Acquisition Myths

One Size Does Not Fit All

CAPT. DAN WARD, USAF

You have probably heard the ancient story of Icarus and Daedalus—how they built wings with feathers and wax to escape the Labyrinth and how Icarus ignored his father's warning about flying too high. Everyone knows that story is a myth.

You have probably heard the following six stories too. The difference is, some people believe these stories are true—and that can lead to serious trouble. Like the feathers in Icarus' wings, the Acquisition Myths described here will ultimately fail to support you as you fly toward your goal.

I THE MYTH OF THE METHOD: "ONE SIZE FITS ALL."

First there was Scientific Management. Later, we had Management by Objectives (MBO), Total Quality Management (TQM), Management By Walking Around (MBWA), The Revolution In Military Affairs, Acquisition Reform, and a host of others. Each method had its particular strengths, and each was rejected, or even vilified, when a new school of thought entered the arena. Today the hot topics include Spiral Acquisition and Agile Acquisition. Tomorrow is sure to bring something new.

For some strange reason, some people tend to get on the TQM/MBO/MBWA/Spiral/etc., bandwagon and become con-

Ward is stationed at the National Imagery and Mapping Agency, Reston, Va. He is the Contracting Officer's Technical Representative for a tactical imagery dissemination system called BRITE. He is Level I-certified in Test and Evaluation and in Program Management, and Level III-certified in Systems Planning, Research, Development and Engineering.



vinced that it will work in every situation, despite the fact that *none of our previous methods were foolproof or flawless*. We humans are hardwired to look for patterns, so why we continue to miss this pattern is a mystery.

The truth is no particular method is appropriate for every conceivable situation, program, or enterprise. To put it



plainly, one size does not fit all. These management methods can be useful and effective tools when applied to the situation they were designed to address, but they quickly become useless or counterproductive when misapplied—and it's not hard to misapply them. When we believe we've got the perfect method, it becomes a box instead of a guide, and we start doing things because

Whether or not Icarus actually plunged into the Mediterranean Sea one sunny day, his story has an important lesson for today's acquisition community. His death is a metaphor for the danger of pride—he trusted his own judgment and flew too close to the sun. It wasn't exuberance or carelessness that brought Icarus down. It was arrogance.

“we're following the method” as opposed to “because it helps us reach our goals.” This also transfers responsibility for failure away from the individual and onto the method. If I'm doing it by the book and something goes wrong, it must be the book's fault!

That is not to say *methods* are a waste of time. Some of them are quite good. But none of them are perfectly suited for every situation—and none ever will be. A hammer is a wonderful tool, unless you need to cut wood. For that, you'll want a saw.

How to proceed? Keep in mind that any method, formula, or process has its strengths and weaknesses. Avoid taking a broad-brush approach to your development efforts, and don't be too quick to latch onto the latest management fad.

2
**THE MYTH OF THE
INTERIM SOLUTION:
“THIS SYSTEM WON'T LAST LONG.”**

Belief in the Myth of the Interim Solution can be dangerous and needlessly expensive in the long run. It can lead a Program Manager to take shortcuts and make decisions that negatively impact the user, because “it is only an interim solution. We'll do the heavy lifting later.” The reality is, if something works, it tends to stick around.

I once worked on a program that actually had the word “interim” in its name. That was several years ago, and as far as I know *it is still in use*. The problem was, the darned thing worked! It met the user's need, inexpensively and simply—so developing the real solution was put off and its funding was diverted elsewhere. Fortunately, we did it right the first time, and our stop-gap capability became a real asset. The truth is, it wasn't a problem at all—other than the fact that we were still calling it an interim solution.

How to proceed? Gold-plating every system is not the answer. Some systems are truly temporary and disposable—just not very many of them. Program Managers and designers need to keep in

MYTH vs. REALITY

- | | |
|---------------------------------------|--|
| 1. One size fits all. | 1. Different situations require different tools. |
| 2. This system won't last long. | 2. If it works, it stays. |
| 3. Requirements creep is bad. | 3. Requirements creep is inevitable and good. Plan for it. |
| 4. We Know the Concept of Operations. | 4. Users are creative and innovative. |
| 5. Development is for the pro's. | 5. Hobby shops can provide excellent systems. |
| 6. We can learn from experience. | 6. We seldom see the long-term results of our actions. |

mind that a system's lifespan will most likely exceed our expectations. We need to be cautious about cutting corners and taking a "we'll fix it later" attitude. In a "pay me now or pay me later" scenario, the upfront payment is often significantly smaller than the bill you'll receive down the road. In other words, it's usually better and cheaper to do it right the first time than to do it over.

The point is, systems sometimes remain operational longer than the developers expect. Just look at the 40-year-old B-52, which is projected to remain in operation until 2037 (or longer?). I wonder how its development might have been different (and how much money could have been saved) if we'd suspected how long it would last.

3

THE MYTH OF REQUIREMENTS CREEP: "REQUIREMENTS CREEP IS BAD AND AVOIDABLE."

Some acquisition programs manage to avoid any significant requirements creep through a variety of approaches. Not that they *should*, but some do.

Here is why a little "creep" is a good thing. As a development program evolves, technology advances—often in unexpected ways. The more time developers spend with users, the better we understand their needs, wants, and expectations. That combination—in-

creased understanding and improved technology—often leads to the conclusion that the system's requirements need to be changed or expanded. That is not a bad thing, and it shouldn't come as a surprise.

This type of requirements creep is therefore largely unavoidable and highly appropriate. If both technology and our understanding of users needs improve over time, it makes sense to count on and plan for the inevitable expansion of system requirements. Problems arise when users recognize the need to expand or change the requirements, while the developer remains focused on the original baseline. Another problem surfaces when new requirements are added without a corresponding increase in funding. These problems are largely self-inflicted and essentially avoidable.

The Spiral Acquisition model is well suited to solve these problems, and it does so admirably. Users receive new capabilities sooner than in a traditional approach, and as technology develops, they receive incremental improvements to the capabilities. Ideas that once would have been labeled requirements creep now can be folded into future spirals.

How to proceed? Rather than fighting requirements creep or seeing it as a necessary evil, PMs should smile and include it in their original plans, budgets, and schedules. The Spiral Acquisition

model, while not suited for all situations (see Myth No.1), gives planners, developers, and managers a flexible road map for such planning. Remember, it is only requirements creep if we didn't see it coming, and there is no good reason to be caught off guard.

4

THE MYTH OF THE CONOPS: "WE KNOW HOW THE SYSTEM WILL BE USED."

While painting my living room walls recently, I used a flathead screwdriver to pry the lid off the paint can. When I was done, I used the butt end of the screwdriver to pound the lid back on. Prying and pounding are outside the scope of a traditional screwdriver CONOPS [concept of operations], and those are not the activities I had in mind when I bought the screwdriver. Still, it got the job done quite nicely, and I don't think I'm the only screwdriver operator who uses it that way.

Warfighters are famous for taking a similar approach to their tools. No matter how experienced, educated, or intuitive a PM might be, we can seldom foresee all the ways our systems will be used and changed after they are deployed. New situations arise unforeseen, and innovative people play with the equipment and make it do new things. Before long, the original CONOPS becomes at best incomplete and at worst obsolete. For example, the American fighter jets still fly, but these days they spend a lot more time doing air-to-ground missions than engaging in actual dogfights with enemy fighter jets. I'm not sure that was the original plan, but there's nothing wrong with that. It is better to have an obsolete CONOPS and a new capability than a solid, unchangeable CONOPS and no innovation.

How to proceed? Aim to produce systems that are adaptable, flexible, scalable, reusable, modular, and interoperable. Keep in mind that users are creative. We don't always know everything about how they use today's systems, let alone how they might use tomorrow's. When you're defining the specs and requirements for the next gen-

eration screwdriver, try to ensure it won't accidentally lose any "non-spec" capabilities (like opening paint cans). Ever try prying open a can with a Phillips head screwdriver? Sometimes losing a capability like that can be an acceptable trade-off. Nevertheless, it should be an *intentional* trade-off, made with the full knowledge (and preferably the buy-in) of the users.

5

**THE MYTH OF HOBBY SHOPS:
"ONLY 'PROFESSIONALS'
SHOULD DEVELOP SYSTEMS."**

At a recent gathering of military acquisition professionals, someone bemoaned the fact that "hobby shops" are "[providing] near-term solutions with no integration, and the folks who own those 'hobby shops' are reluctant to yield any control [over the system]... because they work." It is interesting that anyone would complain about small groups of people providing systems that work. It is even more interesting that the proposed solution is to hand control over to a second party, who probably has less knowledge about the mission need, the system, and the CONOPS than the original hobby shoppers.

The truth is users and other amateur developers are sometimes able to create systems that work quite nicely thank you very much, and they often do it faster and cheaper than anyone else. They may not know their EVM [Earned Value Management] from a hole in the ground, but they know what their operational requirements are. The problem is their solutions might be too tailored to their own situation, so the rest of us miss out on sharing their accomplishment. Their lack of development experience might cause them to make mistakes that a Level III acquisition professional wouldn't. This is exactly where the professionals should come in. Rather than trying to wrest control away from successful amateurs, we should come alongside them and share our professional expertise. There is a word for that type of behavior—teamwork.

How to proceed? As Chief of Staff of the Air Force Gen. John Jumper recently

pointed out, "there needs to be oversight and standards, not standardization." In other words, standards are tremendously useful and important, but standardization misses the point. Integration is often a vital requirement, except when it isn't. What we often lack is not external control over hobby shops, but clearly defined and well-understood standards. The professional acquisition community should let the hobby shoppers keep doing their thing—dreaming up and developing systems that work. Our job should be to join their teams and help them understand that wax melts if they fly too close to the sun, and feathers get heavy if they fly too close to the sea. Maybe they could try using aluminum.

6

**THE MYTH OF LEARNING
FROM EXPERIENCE:
"WE DIRECTLY EXPERIENCE
OR OBSERVE THE
CONSEQUENCES OF OUR ACTIONS."**

Experience is an excellent teacher—perhaps the best teacher around. That doesn't mean we always learn the right things under its guidance. In today's defense acquisition environment, we often have five-year development programs managed by people on two-year assignments, spending one-year money. Military personnel like myself, often can't stick around long enough to observe firsthand the long-term outcomes of our decisions. Experience is an excellent teacher, but it is hard to learn from experience if you're not there.

Peter Senge made this same point in his book *The Fifth Discipline*. As he explains what he calls "the delusion of learning from experience," he points out that we do not directly experience the long-term consequences of many of our most important decisions. Cause and effect are not closely related in time and space, making it nearly impossible to draw proper conclusions and learn proper lessons. He also points out that many of today's problems come from yesterday's "solutions." Why? Those solutions are often based on things we think we learned from experience. We are learning from experience all right, but some-

times we're learning the wrong lessons because our experiences are not always what we think they are.

How to proceed? We can learn from other people's experiences, we can study history, and we can seek out the decisions and actions of past years, watching for causal relationships with today's lessons and challenges. We should also recognize the role that intuition, insight, introspection, and innovational urges can play. Remember, it is very difficult to directly observe all the implications of our own actions. For that reason, it is important to cast a jaded eye on the short-term conclusions we are tempted to draw. Keep in mind that the final chapter has yet to be written.

Acknowledging Our Limitations

Whether or not Icarus actually plunged into the Mediterranean Sea one sunny day, his story has an important lesson for today's acquisition community. His death is a metaphor for the danger of pride—he trusted his own judgment instead of listening to his father's wise counsel. He flew too close to the sun. It wasn't exuberance or carelessness that brought Icarus down. It was arrogance. And arrogance lies at the core of these six myths as well.

If we want to avoid sharing Icarus' fate, we need to steer clear of his flight path. In contrast to the myths described here, the truth is we have not discovered the perfect management or acquisition method, and we never will. We do not know everything about how our systems will be used or how long they will last. We have not defined all our requirements perfectly up-front, and we do not directly experience the consequences of some of our most important decisions. We may be highly educated and highly experienced, but we are also highly human. If we're not humble enough to acknowledge our limitations and smart enough to act accordingly, we will probably end up all wet, just like Icarus.

Editor's Note: The author welcomes questions or comments on this article. Contact Ward at WardD@nima.mil.