A Call for Action

Lt. Col. Dan Ward, USAF

Hi! I want to talk directly to you, dear readers, particularly those I haven't corresponded with personally or haven't actually met. See, I've got a mission for you. I need your help.

For years, I've been writing about ways to improve the outcomes of our acquisition activities. I've used stories and comics and focused on ideas and principles. I hope that stuff has been both helpful and entertaining—I know I've had fun doing it. This time, I want to really get down to brass tacks and propose some specific actions.

Ward *is the chief of acquisition innovation in the Acquisition Chief Process Office, Office of the Deputy Assistant Secretary of the Air Force for Acquisitions Integration.*

Please don't mistake this as a checklist of 10 easy steps to program management bliss. The things I'm asking you to do are neither easy nor guaranteed to deliver results. They may not help you on your particular project or activity. Nonetheless, I hope you'll do at least some of these things—not because I asked, but because they make sense to you for your particular situation.

Have "The Talk"

Regular readers already know I write about values a lot. When I use that word, I'm talking about values as preferences and priorities, not ethics and morals. In this context, values are the measures of merit, the signs of sophistication that indicate whether we've done good or not. Lately, I've taken to calling values meta-requirements because they are the means by which we judge the validity and worth of other requirements within the system, function, organization, or process.

It's important to be deliberate about our values. If we're not, we end up being propelled by the unconscious inertia of invisible values ... which may or may not be constructive. And along with understanding our own values, we also need to be aware of our teammates' values and priorities. When we are unaware of the different values held by various partners, we tend to encounter unproductive friction.

So the first thing I'm asking you to do is talk about values with the people around you—the contractors, customers, senior leaders, and engineers on the project. Getting started is as easy as asking a few questions. You might begin by asking "What is the most important aspect of this system/organization/process/briefing?" You're looking for answers that address things like timeline, cost, complexity, and size. You could cut right to the quick and ask, "Is it important that we deliver this on time? On budget? Or are delays and cost increases acceptable if they result in a bigger, shinier, more advanced system?" The answers might surprise you.

It could be interesting to ask questions like "Would it be acceptable to deliver 70 percent of the capability if we did it for 50 percent of the time and cost?" If the answer is yes, you know the project leaders value being fast and inexpensive. If the answer is no, then the project leaders clearly value something else, like delivering a 100 percent solution in response to a stated need.

Regular readers know I think certain sets of values are more productive and appropriate than others. I'm quite fond of a value set called FIST (Fast, Inexpensive, Simple, Tiny), and I offer it for your consideration. But regardless of what values your team embraces, the first step is to discover what the team's values actually are. So please, have the values talk.

One more thing—while I refer to *the* values talk, it is not actually a one-time event. It's more of an ongoing conversation. You might want to write some of your discussions down and refer to them at key decision points later. That way, when you hear, "We're going to slip the schedule so the technology has more time to mature," you can reply "Really? I thought we'd agreed it was important to deliver quickly. ..."

What is the most important aspect of this system/organization/process/briefing?

And by the way, it's never too late to start the values conversation.

Be Fast and Incremental

I want you to set requirements that can be satisfied in a short timeframe. That's entirely consistent with DoD's overarching acquisition policy and guidance, if not our general practice. As a rule of thumb, I'd say we should aim for less than two years from conception to initial operating capability (DoD and the Government Accountability Office say less than five). In some cases (i.e., certain software development efforts) two years is w-a-a-a-y too long. In other cases, it's a bit on the aggressive side, but for the most part, I think two years is a good target, precisely because it's aggressive. Please don't get too wrapped up in endless debates over when to start or stop the clock; how we define "timeline;" or whether the maximum should be one, two, or five years. The important thing is to deliver systems quickly, however you measure it in your particular context.

You may not be the one who writes the requirements, but if you have any role in shaping, documenting, expressing, or interpreting them, you have an opportunity to push them in the direction of short timelines. I recommend this because I value being fast—and I think it's a productive value for a wide range of system development projects.

Maybe speed isn't something your team values, but it probably should be. A recent briefing by the Zachary Lemnios, director of defense research and engineering, quoted several value-rich statements from the combatant commanders such as "I need the 70 percent solution today rather than the 100 percent solution in five to eight years," and "I like the one-year acquisition cycle rather than the standard five to eight year cycle." Those statements are profound expressions from our customers of the importance of speed. They clearly point to being fast as a meta-requirement that should shape the development and interpretation of subsequent requirements. If you think your team doesn't need to value speed, make sure you confirm that with your customers.

Sometimes it is hard to make decisions that express the speed value. If the combatant commanders quoted in the previous paragraph are among your customers, you're in luck. They've already told us they value being fast and want things to move quickly. You'll probably get good support in your efforts to be fast. My experience with customers, however, is that they're really excited about getting something developed and delivered on a short timeline, but sometimes we have to remind them they don't really want to wait for the 100 percent solution. This gets a lot easier if you're already having the values conversation.

On that note, it's important for everyone to understand we are not simply choosing between a partial or a complete solution. It's actually a choice between a partial solution or no solution at all. That is, today's 70 percent solution has real value for the current fight while tomorrow's 100 percent solution does not.

It is also important to remember that an incremental strategy delivers a 70 percent solution now, an 80 percent solution next, and so on as opposed to supposedly delivering a hypothetical 100 percent solution ~~five~~, ~~six~~, ~~seven~~, ~~eight~~, nine years from now. This iterative approach has the added benefit of ensuring our systems are operationally relevant and technically up-to-date. And isn't that the mission of acquisitions—to deliver affordable operational systems that are available when needed and effective when used?

So I'm asking you to fight like hell to prevent schedule extensions. Do whatever it takes to avoid slipping a milestone—descope the program or shift requirements to a subsequent increment, spiral, or block (pick your favorite term). Please don't slip the current increment's delivery date. As GAO's Director of Acquisition Sourcing and Management Division Paul Francis recommends, "Allow schedule to constrain the design." Again, this is much easier to do if you've already started the values conversation. Whatever you do, never extend your schedule "to let the technology mature." Build operational systems out of technology that is already mature. Trust me, there is always a large body of mature, underused technology just waiting to be sent into action.

Today's 70 percent solution has real value for the current fight while tomorrow's 100 percent solution does not.

Be Cheap and Flexible

If it's at all possible, avoid using the "Here's what I absolutely need the system to do, how much is it going to cost?" approach. Instead, frame the scenario as "Here's how much money I have, how much capability can I get?" To state it more formally, use fixed cost and floating requirements instead of fixed requirements and floating costs. Sure, some people will still promise the moon in response to this situation, but when the inevitable problems arise, you will already be postured to adjust the requirements instead of extending the schedule and budget. So along with allowing schedules to constrain the design, I'm asking you to allow budgets to constrain the design as well.

The underlying idea is that it's better to deliver something useful now than to promise something useful later. This is another case where using mature technologies pays dividends. Because a mature technology is a known quantity, we can produce more reliable schedules and budgets. We get less instability because there's more knowledge. And for the data-inclined, the aforementioned assessment by Francis shows the cost growth of research, development, test, and evaluation programs using immature technologies is orders of magnitude larger than those using mature tech. Google® it, or send me an e-mail and I'll hook you up with his actual briefing.

Exercise Restraint

Ultimately, I am asking you to allow both the schedule and the budget to constrain the design. That's what Francis is asking too. I know this can be difficult. As an engineer, I am fully aware of the temptation to improve a system by adding new widgets. It's what engineers do. Adding components and functions is a sign that we made a contribution to the design, an indication that we've done some work and earned our pay. But good engineers know the real work, the most valuable work, always comes down to simplifying the design, stripping away the extraneous in order to reveal the essential. And good engineers know that delivering the system is the ultimate measure of success. Restraint increases the likelihood of delivering something useful in an operationally relevant timeline.

This points us again to the issue of values. Do we value simplicity in the system, or is it more important to provide a hundred different functions and components? Do we recognize that an elegant, simple design is evidence of deep thought and much effort, or do we only see signs of achievement in complexity? Do we trust simplicity? Or do we prefer complexity? This is an important topic for program managers to discuss with the engineers and customers because it gets to the core of what constitutes good design and good work.

NASA program managers on the Near Earth Asteroid Rendezvous (NEAR) mission deliberately resisted incorporating "good ideas" into the system while still acknowledging that the ideas were good. That design restraint expressed appreciation for the people who expressed the "good ideas" and also avoided increases to the mission's cost, schedule and complexity. Readers familiar with my "Faster, Better, Cheaper Revisited" article (*Defense AT&L*, March-April 2010) already know the details of this highly successful mission. For now, I will simply reiterate that their success was the result of firm, values-driven restraint that focused on delivering the essential capability. It is exactly this kind of productive, creative restraint I hope you will exercise on your project.

Read Good Books

Since you're reading this article, I assume you place some value on reading in general. No doubt you already make time to read other things, and if you're like me, you probably have a perpetually growing stack of books to read someday. At the risk of making your reading list even longer, I'm recommending a few titles to consider (see sidebar). But whether you read these books or some others, keep reading good books. Read the really good ones again.

Share Your Story

Share your story, and there are numerous ways to do that. You can write your story, but trust me on this one; writing is hard work. It's time consuming and often frustrating. But when it works, it's also a lot of fun. So I want to encourage you to write something and get it published. I'll bet you have an opinion on something related to defense acquisitions, an experience worth reflecting on, a lesson worth sharing. Maybe the only reason you haven't put it down on paper yet is because nobody asked you to. So I'm asking. If you don't tell your story, who will?

Alternatively, you could send me an e-mail at <daniel.ward@pentagon.af.mil>. I really want to hear from you. I want to hear your stories, your triumphs, and your trials. I'd love to field your questions and receive your critiques. I'm more than happy to be a sounding board and would love to get together over coffee if our geospatial coordinates intersect. But whether you write to me or not, I hope you'll tell your stories to someone. Grab a buddy and go out to lunch. Write to an old boss, professor, or colleague. Talk about your projects, past or present. Reflect on the way your values shaped your decision making. It's time well spent.

Books To Read

- *Boyd*, by Robert Coram
- *The Chaordic Age*, by Dee Hock
- *Orbiting the Giant Hairball*, by Gordon MacKenzie
- *Maverick*, by Ricardo Semler
- *Re-Imagine*, by Tom Peters
- *Losing My Virginity*, by Richard Branson
- *The Reflective Practitioner*, by Donald Schron
- *The Hypomanic Edge*, by John Gartner
- *The Hacker Ethic*, by Pekka Himanen
- *Up The Organization*, by Robert Townsend

Connect

One of the great things about writing for *Defense AT&L* is the opportunity to hear from readers. It's fun to share ideas and stories with so many of you, to commiserate and celebrate life in the defense acquisition community. So the next thing I'm asking you to do is connect with each other.

I think it would be great to have an online forum where readers can connect, share, and learn. This place would have links to resources (articles, briefings, conferences, etc), and discussion threads on various topics. I know there are several in-house platforms out there that provide this kind of capability as well as plenty of commercial platforms, but I'm not sure we've really gained critical mass on any particular one. I'd love to hear your thoughts and suggestions on this. If you're already using one, I hope you'll send me an invitation to join in. If one of you has a better way to connect, I hope you'll share your solution.

Share This Article With Someone

Here's something everyone can do: share this article with someone. Share it with your team, your boss, or your customer. I hope this request doesn't sound self-serving. I'm just as happy to have you poke holes in my ideas, debate them, or challenge them as to recommend or defend them. But mostly, I hope you can use this article to help start a discussion about your team's values and how they shape decisions and behavior on the project.

Finally...

If this is your first time reading one of my articles, welcome and thanks for taking the time to read. I hope you found something useful here, and I invite you to check *Defense AT&L*'s online archive (<http://www.dau.mil/pubscats/pages/damtoc_new.aspx>) for a wide range of previous articles by oodles of other writers. Good luck out there. Take care of each other. And if you have a moment, drop me a line.

The author welcomes comments and questions and can be contacted at **daniel.ward@pentagon.af.mil**.