

# Requirements: The Root Of All Evil

## Observations on Life, Death, and Requirement Definition in Defense Acquisitions

*Capt. Daniel Ward, USAF*

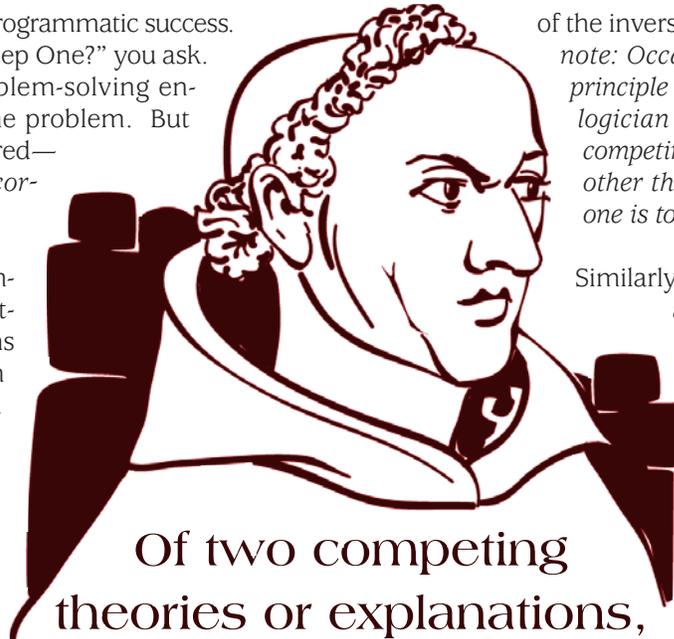
**S**tep Two is the key to programmatic success. “What happened to Step One?” you ask. Step One in any problem-solving endeavor is to define the problem. But Step Two—often ignored—is to define the problem *correctly*.

Every development effort inside or outside the Department of Defense (DoD) begins with some form of problem statement. Whether it’s a mission needs statement, operational requirements document, statement of objectives, or a combination thereof, at some point program managers are faced with what can generically be called a requirement—a description of something that someone needs.

Since this starting point largely drives all subsequent tasks, it’s very important to get it right. And since operators and technologists tend to speak different languages, getting it right can be very difficult. Requirements are, therefore, the lingua franca that operators and developers use to establish a common understanding of the operators’ needs and the developers’ intentions. Like any language, proficiency in “requirement speak” comes with study, practice and prolonged exposure to native speakers. This article, of course, falls in the study category—actual usage is up to the reader.

### Occam Undone

As H. L. Mencken said, “For every human problem, there is a neat, simple solution; and it is always wrong.” Sort



Of two competing theories or explanations, all other things being equal, the simpler one is to be preferred.

of the inverse of Occam’s Razor. [Editor’s note: Occam’s or Ockham’s Razor is a principle attributed to the 14th century logician William of Ockham: Of two competing theories or explanations, all other things being equal, the simpler one is to be preferred.]

Similarly, for every situation, there is a problem statement which is obvious, simple—and likely to be absolutely incorrect. It isn’t that simplicity plus obviousness always equal the wrong answer. After all, good solutions often are obvious and simple. But the point is that not every obvious and simple “solution” is a good one. The reason so many problem statements are bad is they not only presuppose a solution, but they settle for the obvious/simple/wrong solution.

But by talking about solutions, we are getting ahead of ourselves. Real solutions to real problems are much easier to find if the actual problem is well understood and clearly stated, without presupposing any particular solution. The problem statement, therefore, must be simple but is seldom obvious except in retrospect. Hence the need for Step Two, since our first attempt is often incomplete or incorrect.

### The Only Thing You Have To Do

My seventh grade math teacher, Mr. Byther, always gave the same answer when we asked if we had to do our homework in a particular way. In fact, he sometimes gave this answer even if we didn’t ask. With a broad grin,

*Ward is an InnoVisioneer in the Horizontal Integration Office of the National Imagery and Mapping Agency’s InnoVision Directorate. He is Level-III certified in SPRDE and Level-I certified in PM and T&E.*

he told us, “The only thing you *have* to do is die.” He mathematically divided the famous certainties of death and taxes in half, and death was the only remainder. His point was this: there is only one real requirement. You can always live in the woods and avoid paying taxes or refuse to pay and get sent to jail, but eventually we all meet our maker. Mr. Byther’s ability to get to the heart of the problem—the real requirement—has been a lasting lesson these many years since seventh grade.

### What A Tangled Web We Weave

“The truth,” as Oscar Wilde put it, “is rarely pure and never simple.”

#### Example Number One: We Need More Analysts!

A frequent complaint in the Intelligence Community (IC) is the shortage of analysts, and no doubt there is a need for more people with the rare and valuable skills necessary to interpret and understand the vast quantities of intelligence data collected every day. However, anecdotal observations lead to a somewhat modified problem statement.

Watching a number of analysts at work, it becomes apparent that they spend a lot of time trying to collect and access relevant data and relatively little time doing actual analysis. In fact, some analysts estimate that upwards of 50 percent of their time is spent searching for data. So perhaps the problem is not merely too few analysts but too much difficulty accessing data.

Why is it so difficult to access data? One possible reason is that the tools provided by the acquisition and technology community are too difficult to use. Why weren’t simpler tools provided? Perhaps because the analysts didn’t submit a requirement for them. Why didn’t the analysts submit a requirement? Perhaps because the requirement submission process is too difficult and mysterious. Or perhaps because they subscribed to received wisdom that

the problem was a shortage of analysts and didn’t think any further.

The IC’s problem now sounds quite a bit more convoluted than a simple shortage of analysts—and even so, it’s likely we haven’t defined the problem in terms of a root cause.

Where does it end? It ends when we are finally able to define what the actual problem is, and not in terms that beg the question or presuppose a quick solution. Are there too few analysts? Perhaps. But the *actual* problem is deeper than that, and until the actual problem is identified, it will probably not be resolved.

While we shouldn’t undervalue a gut-level assessment of what’s needed, we can’t simply stop there either. Our understanding of a problem drives the requirements we levy in an attempt to solve the problem. That is why it is important to understand the actual problem and to write requirements that do not dictate solutions.

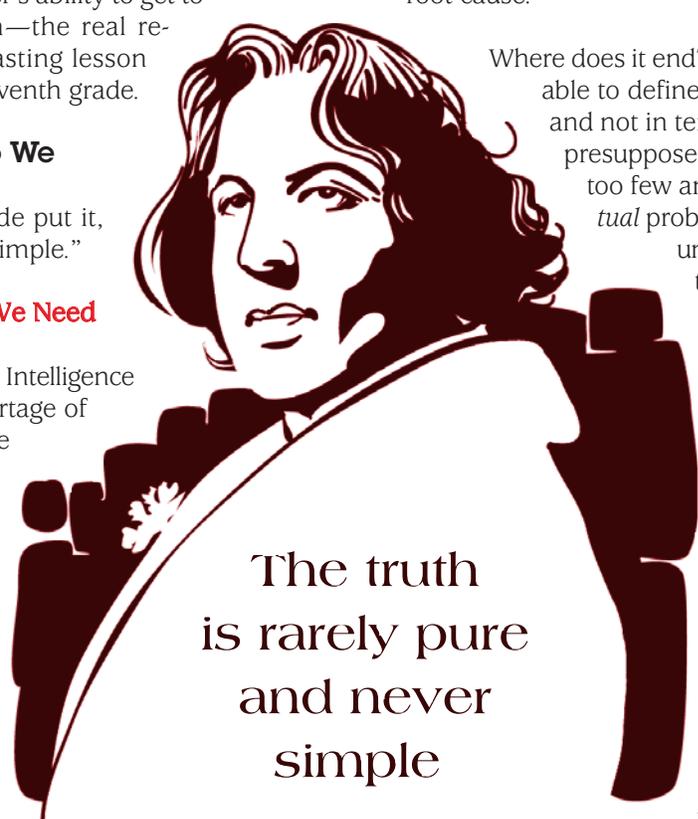
Warfighters don’t need System X. They need to be able to do A, B, and C.

#### Example Number Two: We Need More Training!

At a recent conference, loud complaints were voiced about a lack of training for particular operational specialties. However, “We need more training!” is a problem statement that presupposes an easy solution. There may indeed be a significant training shortfall, but the root of the problem is deeper and merits a closer look.

Rather than simply needing more training, perhaps these individuals need easier-to-use systems that don’t have such a steep learning curve. Perhaps they need a more focused and consistent mission or a decreased rate of personnel turnover. Maybe they just need encouragement and appreciation.

Most likely, the requirement is something along the lines of this: “We need to produce a specific effect. Producing this effect given current rates of personnel transfer and with our current systems requires a larger training investment than we are currently making. So, we either need more or better training, less frequent rotations, easier-to-use systems, or a simplified mission.” This state-



### A Good Requirement

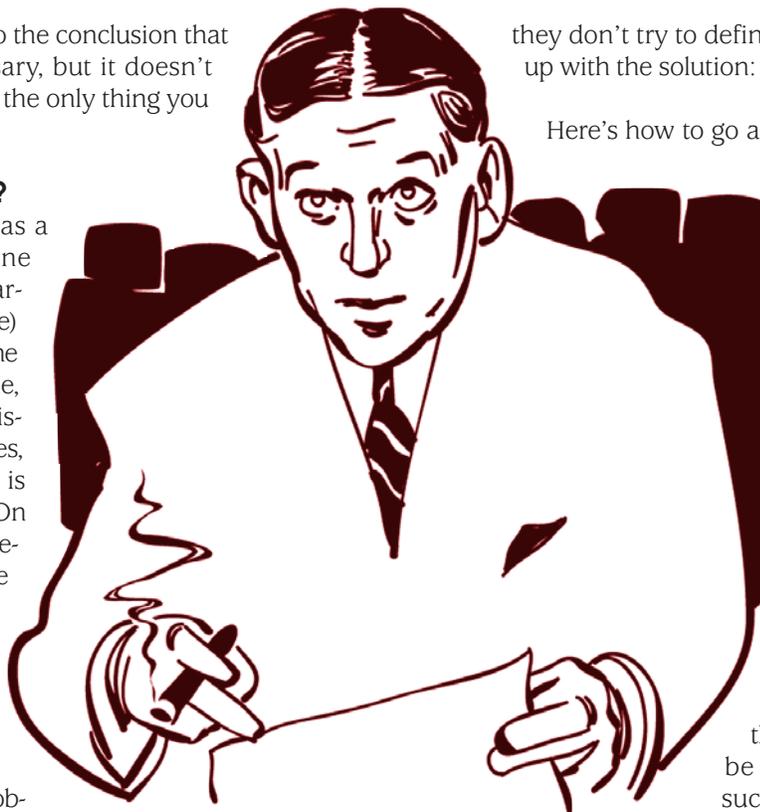
- Is measurable and achievable
- Doesn't get written in stone
- Doesn't presuppose a solution
- Can be met in a variety of ways

ment may indeed lead to the conclusion that more training is necessary, but it doesn't start there. (Remember, the only thing you *have* to do is die.)

### The Root of All Evil?

The title of this piece has a double meaning. On one hand, requirements (particularly when they change) are occasionally seen as the program manager's bane, despite the fact that satisfying the requirements (yes, even the changing ones) is the PM's primary task. On the other hand, a good requirement identifies the root cause of some type of "evil"—an operational shortfall currently unmet by existing capabilities—and so a well-done requirement is indeed the core of a problem. Although we may not like to admit it, defining a requirement correctly often takes more than one attempt.

How then shall we proceed? At the risk of contradicting the esteemed Dr. Steven Covey, highly effective people do not start with the end in mind. That is,



For every human problem  
there is a neat, simple  
solution; and it is  
always wrong.

they don't try to define the problem by coming up with the solution: "We need more training!"

Here's how to go about it. Investigate deeply and identify the desired capability or effect, which may be arrived at by a number of paths, including some that are as yet undefined. Connect developers and operators early and often, to ensure they are all speaking the same language. Take the advice of software guru Eric Raymond and "expect to start over at least once." Establishing a mechanism for implementing Step Two (define the problem correctly) may be difficult, but not having such a mechanism is unacceptably foolish. It gets easier if we admit the existence and validity of changing requirements and accept the fact that a target does not cease to be a target when it starts moving.

**Editor's note:** The author welcomes comments and suggestions. He can be reached at [wardd@nima.mil](mailto:wardd@nima.mil).

## The American Soldier

**T**he American Soldier, representing all men and women who wear the uniform, has been named Time magazine's Person of the Year. Three 1st Armored Division soldiers—Sgt. Ronald Buxton, Spc. Billie Grimes and Sgt. Marquette Whiteside—are featured on the magazine's Dec. 29 – Jan. 5 cover.

Photo by James Nachtwey/VII for *Time*, used by permission

