Instar

Doing Things Right in Space Programs

This article is part of a series started in January, 2000. My intent is to share a philosophy and ideas for how to increase the chances of success in space missions while also reducing total cost. Once these articles are completed, I plan to assemble them into a book. Please send comments to me at Tom.Sarafin@instarengineering.com.

Ten Principles for Doing Things Right in Space Programs

1. **Adopt the right attitude**
2. **Invest in knowledge and understanding**
3. **Instill ownership and responsibility**
4. **Constantly seek ways to improve teamwork**
5. **Follow a sound engineering approach**
6. **Reduce total cost through good engineering**
7. **Keep everything as simple as possible**
8. **Establish an effective quality system that involves everyone**
9. **Be willing to accept risks, but only those you truly understand**
10. **Make sure everyone has enough time, resources, and freedom to do things right**

Article #8

Instill Ownership and Responsibility

August 2000

Revised October 2002

Tom Sarafin

President, Instar Engineering and Consulting, Inc.

6901 S. Pierce St., Suite 384, Littleton, CO 80128 • (303) 973-2316 • instarengineering.com

Ensuring quality products and work from employees and contractors is a challenging problem. Many contractors seem to want to do as little as possible to satisfy requirements, with little care for the quality of their products. More disturbingly, some of the people within our own organization may show a similar attitude. We don't like it, but we don't know how to change it.

I'd like to share some thoughts on this matter. The goals here are to understand why things are the way they are and to create an environment that enables and encourages ownership and responsibility. The topics I'll address are how we specify requirements, how we award contracts, how we insure space missions. I'll then discuss things we can do to improve the attitude and increase the sense of personal responsibility on the part of the people within our own organization.

Improving How We Specify Requirements

A common attempt to solve the problem of how to ensure quality from employees and suppliers is to specify requirements or criteria related to everything we can think of that could affect quality. We write hundreds of standards and procedures, and we fill product specifications with required tests, test margins, analyses, factors of safety, material inspections, and on and on. This approach doesn't work.

I first began to realize that this approach doesn't work in the late 1980's, when I was leading a group of seventeen stress analysts at Martin Marietta. Of this group, I trusted the judgment of two or three. Not that they knew everything about stress analysis, but I could turn them loose and was confident that they would make sound decisions, including that of alerting me if they got in over their heads. The rest of my people were relatively inexperienced, and I hadn't developed confidence in their judgment.

I didn't know how to handle this problem. I believed in the principle of empowerment, but not if it risked meeting our group's responsibility to the program. If the spacecraft structure failed, the entire mission would fail, or someone might die. Education is what my group needed, but most of us were already working overtime because of a heavy workload; I couldn't find any suitable educational material, and I didn't have time to develop it myself.

Then the solution hit me: I decided to write and distribute a structural design criteria document. So I sat in my office and composed a 40-page document, drawing on similar ones from other programs and supplementing them. For everything I could think of that could affect structural integrity, I wrote a rule. When I was finished, I carried it down the mountain and laid a copy on everyone's desk. Then I went home and slept like a baby, figuring my problem was solved.

A couple weeks later, I noticed one of my people—Rick—sitting at his desk with his head in his hands. I asked him what was wrong, and he raised his head and, with a heavy sigh, said, "My head hurts. I have too many requirements. I can't keep them all straight!"

I can't say that the truth hit me at that moment like a ton of bricks, but the light bulb did begin to flicker for the first time. (Since then, the bulb has burned much brighter. Later in this article, I'll share an experience that more than doubled the wattage.) I first started to see that you can't specify solutions to problems that people don't understand. I had recognized that Rick didn't understand the technical aspects of his job well enough, but I took the easy approach to fixing that problem. I needed to find some way to help him and the others understand the problem better, perhaps through lunch-time group sessions in which we discussed our responsibility, the kinds of things that could go wrong, and what we'd do to make sure they wouldn't. We could have then written down our conclusions in the form of a criteria document that everyone not only understood but also owned.

We have a similar problem in our dealings with contractors, often on a much larger scale. We tend to trust contractors far less than we trust our own people, and so we fill our specifications with everything we can think of to ensure our contractors' products will do what they're supposed to do. Our contractors might not recognize that, before committing to production of an expensive product, there should be analysis showing the product should work, so we'll require analysis and specify the factor of safety and thermal margin. Their analysis might be wrong, so we'll specify a test. They might not understand that a test is supposed to envelop the peak environment expected during the mission, so we'll specify the test environment for them. They might not understand that materials can have flaws, so we'll specify the need for inspections.

Believe me: if your contractors are idiots, they'll screw things up no matter how many quality requirements you put in the spec.

Even if they're not idiots (usually the case), the more verification and quality activities you specify, the more your contractors will act like idiots. They'll try to memorize a slew of requirements (like Rick), and treat the specification like a checklist: if it's not on the list, it won't get done. They'll do a test because the spec tells them to, not because it builds confidence in any way; and they'll do the cheapest test possible to meet the spec, even if the test adds no true value.

Think of what I'm saying here from both perspectives, as the contractor and as the customer. Many of us in this industry have been both.

Let's assume first that you are the contractor. Your customer tells you in a specification not only all the analyses, inspections, and tests you must do, but also all the ground rules for ensuring those activities will be legitimate or valuable. Your first thoughts may be, "Great! They've saved us a lot of work; we'll hardly have to think. After all, they're the customer, and the customer is always right. We'll provide the best service we can and do what the customer wants."

There is a higher level of customer service than that! The only reason your customer is telling you to do all those things is that your customer doesn't trust you! Is this the kind of relationship you want?

Now look at it from the customer's point of view. You don't trust your contractor; not that they've done slimy things, it's just that they haven't yet earned your trust. So you load up the spec with verification and quality requirements—everything you can think of. They do all the analyses, inspections, and tests you specified, but, after taking delivery (or after launch), you find the product doesn't work.

Whose fault is it?

Don't you want your contractor organization to feel that it would be their fault? That it's their responsibility for ensuring their product works **during the mission** rather than simply checking off requirements on a list?

The more complex the product you are procuring becomes, the more you need the contractor to have that sense of responsibility. After all, it's their design. Who will know best how to ensure that design will work? You can't expect to anticipate everything for a design you hardly understand (and that probably hasn't been completed).

When we specify requirements, we need to remember to specify requirements only, and not verification or quality assurance.

- A true *requirement* is something the product must be able to do, or some characteristic the product must have.
  - Failure to meet a true requirement implies mission failure or reduced performance.
- A *verification activity* is something intended to demonstrate that the product will meet its requirements.

- *Verification criteria* are ground rules for the verification activity to be considered adequate.
  – Failure to meet a criterion does not mean the mission will fail; it implies greater risk of failure, so it merely drives the need for assessment.

Use these definitions to distinguish between a requirement and a criterion. The former belongs in a specification written and owned by the customer; the latter belongs in a verification and quality-assurance plan written and owned by the contractor. Before there is a contract, both documents must be acceptable to both parties. This means that, even though the contractor owns the verification plan, the contractor must sell it to the customer before the contract is made.

I say "must," here, even though what I'm describing is not standard practice. The common practice of awarding contracts without an adequate, contractor-written verification plan has lead, in my opinion, to many costly disagreements as well as irresponsible contractors and failed missions. Many customers actually do require a contractor-written verification plan, but then they specify in the requirements document what that plan essentially must be. The contractor then simply cuts and pastes the words out of the spec and into the verification plan. There is no value in this.

Requirements stem from two things: *functions* (what the product must be able to do) and *constraints*, such as interfaces and mass restrictions. After writing a requirement, ask yourself, "Is this a function? Is it a legitimate constraint? Or am I specifying how someone must demonstrate that the product will function properly or meet its constraints?"

Read through the requirements listed in Table 8-1, and see if you can identify those that are true requirements, which should be in the specification, and those that are verification criteria, which should be in the verification plan.

Table 8-1. Requirements or Verification? Where do these “requirements” belong: the customer-owned specification or the contractor-owned verification plan?

1. The ORS qualification unit shall be subjected to the qualification-level temperature range specified in Table 3-4.
2. The ORS shall perform as specified after exposure to the random vibration environment specified in Fig. 3-5 for a period of one minute in each of three orthogonal axes.
3. The ORS shall be protected from contamination according to the requirements specified in FED-STD-209.
4. The ORS shall meet the fracture control requirements specified in NASA-STD-5003.
5. The ORS shall have a reliability of at least 0.95 over 7 years normal use.
6. The ORS shall be able to perform as specified after exposure to the acoustic environment in Fig. 3-7 for a period of one minute.

Of this list, only one is a true requirement (what the ORS must be able to do). The others relate to things that help ensure the ORS will do what it’s supposed to do, so they belong in the verification plan. They are the contractor’s responsibility. Let’s review these requirements individually:

1. The words “shall be subjected to” clearly specify a test, as does reference to a qualification unit.

2. Although the test is somewhat better hidden here, it's still being specified. To avoid specifying a test in this case, we would have to replace "shall perform" with "shall be able to perform" or "shall be capable of performing," which enables other possible methods of verification. Also, the words "three orthogonal axes" refer to testing; during launch the ORS will vibrate randomly in all directions simultaneously. The true requirement is for the item to function properly after exposure to launch as part of a system. Any attempt to derive bounding environments at the item's mounting interface encroaches into the world of verification and thus makes us assume some of our contractor's responsibility. Still, we typically define environments this way (e.g., three orthogonal axes) for relatively small components because it's not practical to do otherwise. For larger systems, though, such as complex payloads for spacecraft, it's best to leave the requirement as being able to withstand launch and then include the payload contractor on the team trying to derive environments. At least, this is the case if we will hard-mount the payload to our spacecraft. If we can mount it with an isolation system that will keep the systems from dynamically coupling, then confidently deriving and specifying interface environments becomes more feasible.

3. and 4. Just because a Government standard exists doesn't make it right to specify it to a contractor. Our country has been deleting many standards, and replacing many others with documents providing guidance instead, in recognition that they drive cost and take away ownership. I hate to see collected wisdom destroyed, so I much prefer the idea of guidance documents over elimination. It should be up to the contractor, then, to determine whether to follow that guidance. Remember, though: if you, as customer, aren't comfortable with your potential contractor's plan for verification and quality assurance, don't award them the contract! (More on this later.)

5. Reliability is almost always specified in major procurements, and I think it's a mistake. The requirement is that the product function as needed—not that the product have a 95% chance of functioning as needed! This is clearly verification we are talking about. The customer should specify what the product must be able to do, and the contractor should engineer the process to provide sufficient confidence that the product will do it. Perhaps they'll recommend to their customer that they design the product to have an analytical probability of 95%. This is such a small difference in how we do business, but look at the potential benefits. Instead of the contractor's people going through a great deal of work and generating piles of paper for the purpose of satisfying a required reliability, they focus instead on developing a product that works. If they generate all that analysis and paperwork on their own, it will be of far greater value because they now believe in them! The same argument applies to the common practice of specifying structural factors of safety and positive margins of safety. Anytime we specify criteria to be satisfied by analysis, the contractor, if incompetent or untrustworthy, could manipulate the analysis until the requirement is satisfied. If the only reason we are specifying such things is that we don't trust our contractor, where does this leave us? With high cost and little true value.

6. Finally, a true requirement. In this case, we've done some of the engineering for our contractor by predicting the acoustic environment and specifying a one-minute duration, but there is probably no practical alternative. The words "shall be able" leave the door open for methods of verification other than testing. You may be convinced an acoustic test is necessary, but perhaps your contractor will surprise you with some clever new approach. If your contractor tries to sell you on accepting their product without a test, and you aren't enamored by the cleverness of their proposed approach, fight off the

tendency to pound your fist on the table and demand a test. Use tact and undeniable logic: raise questions they can't answer with their approach until they, too, see the need for a test. If this doesn't work, remember: you don't have to award them the contract. Just don't wait to address this issue until after you've already given the contract to them!

Although the idea of not specifying verification and quality standards may not be new to you, you may be thinking the extent I'm suggesting might be somewhat radical. After all, nearly everyone in the industry generates specifications according to MIL-STD-490A, which says to include a section (4.0) on quality assurance and verification. We can still do that, but I suggest a different approach. Rather than putting a verification matrix in Sec. 4.0, consider specifying instead the need for the contractor to develop an acceptable plan for verification and quality assurance. Table 8-2 shows an example of how we might structure Sec. 4.0 in a specification.

Not only does this approach build ownership, it also keeps everyone focused on the true requirements by clearly separating them from verification activities. Now, when we encounter unanticipated problems, we can better understand the impact. Does the new information imply the mission will fail, or does it imply greater risk? Separating verification from requirements makes it easier to address risks. No longer are we violating a "requirement."

Table 8-2. Suggested Contents of Section 4.0 (Quality Assurance) in a Specification.

4.1	The contractor shall develop a Verification and Quality Assurance Plan (VQAP) that ensures the product will meet the requirements specified in Section 3.0 herein.
4.2	Before the contract for full-scale development will be awarded, the contractor shall develop a preliminary VQAP that meets the approval of the customer.
4.3	As the product design evolves and more information becomes available, the contractor shall revise the VQAP to meet the intent of the preliminary plan, as judged jointly by contractor and customer representatives.
4.4	Throughout the program, the contractor shall keep the customer informed of changes and compliance status regarding the VQAP.

I know the wording in Table 8-2 goes against what we've learned about how to write requirements; the words "that meets the approval of the customer" and "as judged jointly by contractor and customer representatives" appear to invite trouble. We certainly couldn't decide tomorrow to replace Sec. 4.0 with something like this, not after many years of following a more rigid procurement system, and not with the typical relationship between customer and contractor that exists as a result.

The only way the approach I'm suggesting would work is if competent, broadly knowledgeable technical people from both sides work together with trust and teamwork toward a common goal: a cost-effective, quality product (or successful mission). For this approach to work, you, the customer, must not only be willing to listen to alternatives, but also have a sound understanding of the key issues.

I have some experience working this way. On one space program, I had been hired as a consultant by the system customer organization. My job was to represent their interests regarding structures and environments in the development of their spacecraft. The lead structures and environments engineer for the spacecraft prime contractor called me one day to explain a problem his group was having. He explained they had a design for a gimbal mechanism, and they were confident it would be able to withstand the launch environment, based on using similar designs for past missions. But their analysis predicted that the specified random-vibration environment would overload the gimbal's bearings and cause the races to brinell, or dent. If this happened, the gimbal would not run as smoothly as it needed to during the mission. Using larger bearings would drive weight and complexity throughout the gimbal mechanism. As the prime-contractor organization, they had derived this environment themselves, based on vibroacoustic response analysis.

"Tom," he said, "this is a good, simple design, and we really think it could handle launch. We believe the specified environment is not realistic and is too severe. What do you think? Is it a real requirement?"

His customer, XYZ company, had hired me to represent them in such decisions, having little expertise of their own in structures and environments. If I were satisfied, the customer would be satisfied. Such an arrangement led to valuable flexibility here.

"No," I said. "The requirement is that your gimbal withstand launch and then work properly during the mission. You're the ones who derived that environment. If you think it's too severe, what do you propose doing instead?"

His plan was to test the gimbal as part of a larger assembly, driving the assembly with a random environment tailored to achieve selected target response accelerations near the gimbal. He kept me involved throughout; I even helped them tailor the environment. As a result, we found a way to use the best design. The gimbal passed the test and subsequently contributed to a successful mission.

Now, if this gimbal had been subcontracted, my technical counterpart would have specified that environment to the subcontractor. That by itself is not necessarily wrong; I'd be hard pressed to find a better solution. But, if we go that route, we need to have a close working relationship with our subcontractor and make sure we keep derived requirements such as vibration environments flexible. Otherwise, we may end up with an unnecessary test failure or a much heavier and more expensive gimbal than we need.

I'd like to close this discussion with one more example. I once supported a spacecraft program that had made a key early mistake by agreeing contractually that a major subcontractor providing the payload would verify the payload structure by analysis alone. There would be no structural loads testing or vibration testing.

Flying a structure without a test <u>can</u> be viable (or responsible), but usually only if the structure is simple, with direct load paths, and if the structure doesn't have challenging requirements for dimensional stability. In this case, however, we were dealing with an optical sensor, and the structure had to maintain precise alignment of the optics. In addition, the load paths were quite complex. The metal structure contained bolted joints, welds, and bonded joints.

We had set ourselves up for trouble, and it came in spades. I was on the customer team chartered with becoming confident in the ability of the payload structure to maintain alignment throughout the mission. The lead structural engineer for the payload contractor fully automated the analysis with a detailed finite-element model and software used to search for the highest stress. I, on the other hand, was convinced that, if anything in the structure suffered permanent deformation during launch that would be detrimental to performance, it would the joints—and you can't adequately assess joints solely with a finite-element model. Needless to say, we couldn't reach agreement on whether the structural analysis was adequate verification of structural requirements.

Eventually, the program decided to add a vibration test for the flight structure. We all recognized, however, that, to protect the expensive optics from overtest, the test would not exercise all of the structure high enough to envelop the expected launch loads. I had assumed the contractor's engineers would assess the test conditions and measured responses to determine the extent of testing and then, for complete verification of the structure, supplement with analysis in the areas that hadn't been fully tested. I was mistaken. The contractor insisted on planning and performing the test without our input and afterwards could not tell us how much of the structure had been tested.

I was pulling my hair out in frustration, convinced the contractor's engineers were idiots. Each time they attempted to sell off their structure to us, I went away with more concerns and unanswered questions. Finally, they asked me to fly out and explain to them what I wanted them to do to close out the issue. Of course, they were more than ready to charge us for the extra work. The payload contracts manager on our side expressed his concern, cautioning me not to take away our contractor's ownership. I replied, "I'm sorry, Bob. I don't know what else to do; we're not getting anywhere." So I worked out a detailed verification plan, complete with processes and criteria, and then I flew out and reviewed it with them. The response: "That's not good enough, Tom."

"What do you mean?" I asked.

"We want you to tell us which joints you want us to analyze."

Well, at this point the alarms in my head started going off, and my internal light bulb shone so bright it practically blinded me. I realized that, if I did what they asked, I would fall completely in the trap. If they assessed the joints I told them to, and did it the way I told them to, and if their structure still failed to maintain alignment during the mission—it would be my fault, not theirs.

I said, "Never mind," and I grabbed the plan I had worked up and flew home. We eventually worked off the issue, although all of us incurred a great many scars, and we ended up having a successful mission. Of the many lessons I learned from this experience, the two most important probably are to make sure that what you agree to up front is sound and to make sure your contractor has the technical knowledge and understanding to carry the plan out.

In next month's article, I'll continue to explore the principle of creating an environment that instills ownership and responsibility by looking at how we award contracts, how we insure space missions, and how we can improve things at our own organization.

About the Author

Tom Sarafin has been involved in the space industry full time since 1979, at which time he graduated from The Ohio State University with a BS in civil engineering and took a job as a stress analyst at Martin Marietta Astronautics in Denver, Colorado. While at Martin, he was involved with design, analysis, verification planning, and testing on several spacecraft and launch vehicle programs. After contributing to the book *Space Mission Analysis and Design* [Larson and Wertz, editors, first edition published in 1991], he obtained management's support and funding at Martin Marietta for the development of a book on the interdisciplinary development of structures for space missions, and served as principal author and editor for 23 other authors. He left Martin Marietta in 1993 to complete this book, under the guidance of Dr. Wiley Larson at the U.S. Air Force Academy. The result of nearly four years work—*Spacecraft Structures and Mechanisms: From Concept to Launch*—was published in 1995 jointly by Microcosm, Inc., and Kluwer Academic Publishers.

In 1993, Mr. Sarafin formed his own company, Instar Engineering and Consulting, Inc. Once he finished his book, he began providing review and advice as a consultant to space programs. He also developed a short course based on his book and began teaching it throughout the industry. The course has been quite popular, and the business has grown. Now Instar offers a curriculum of courses taught by experienced engineers and continues to add to that curriculum.