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 #15

Follow a Sound Engineering Approach
Part 4: Verification Logic

June, 2001

Tom Sarafin
President, Instar Engineering and Consulting, Inc.
6901 S. Pierce St., Suite 384, Littleton, CO 80128 • (303) 973-2316 • instarengineering.com

A common question on a space program is, “How do we develop an effective verification plan?” We want to ensure a successful mission, of course, but most of us have been on programs in which, in hindsight, many activities provided questionable benefit.

Experience tells me that planning for effective—yet concise—verification and quality assurance isn't difficult if we truly understand the problem. My last article presented verification as part of a sound engineering approach. Let's continue along these lines by looking at the logic flow for verification. If you accept my definition of verification as "establishing confidence," we certainly begin verification during conceptual design, so we'll start there with the logic flow. Notice that, although I'm referring to verification logic, all I'm really describing is sound engineering. The logic I'm describing here applies to hardware development, but much of it pertains to software as well.

Recall that, during conceptual design, we try to find the best design concept that would meet requirements. To assess alternatives, we must estimate their costs, much of which will arise from trying to establish confidence that they would work. Early planning for verification not only enables us to make these estimates, it also is essential to adequately scope the program so we'll be able to do things right. As I've said many times in these articles, doing things right entails anticipating and avoiding problems, and there is no more important time to do this than when we are selecting a design concept.

For a proposed concept, the key question we must address is, "How well can we predict the product's key characteristics?" The answer depends on our manufacturing processes and on our analyses. To predict a product's characteristics, we must be able to predict what the manufacturing processes can achieve at low levels of assembly and then extrapolate with analysis the characteristics at the subsystem and system levels. The key is to recognize when such extrapolation is not valid.

If our design calls for a new manufacturing process, obviously we must develop it. Process development means much more than defining sequential steps. We must identify process requirements, define a process, test it out to see if it meets its requirements, and then iterate as needed. Process requirements can include limits on cost and schedule, but mostly they relate to product characteristics: an average value (e.g., surface roughness, or strength of a bonded joint) and a measure of scatter or variation. Only by building several representative products with the proposed process and then testing them can we determine whether a process is adequate. Again, as I've mentioned earlier in this series of articles, the space industry seldom has the production volume needed to pay for the extent of process development we would like. But, given the impact of finding out late that a process is inadequate, we still must be reasonably sure of our processes before committing to a design.

The X-33 program was recently cancelled after more than $1 billion had been spent on it. The event that probably killed the program, even though it happened more than a year before cancellation, was test failure of the composite propellant tank. Given the ultimate consequence of this failure—death of a billion-dollar program—we have to ask if this flaw, whether in the design or in the manufacturing process, could have been found in early development testing, when failure would have had little impact. An article in the November 15, 1999, issue of *Aviation Week and Space Technology* adds fuel to this question by sharing the opinions of several engineers on the X-33 program that development testing had been inadequate. According to the article, the program manager defended the extent of development testing and criticized a statistical study that indicated the manufacturing process was insufficient. His argument was that such studies are not appropriate for experimental programs. This argument holds no weight with me. Any expenditure of over $1 billion more than justifies a sound engineering process, which, when developing a process, requires supporting statistical data.

The goal of verification during conceptual design is to become confident enough to commit to the concept and proceed with full-scale development. Figure 15-1 shows the verification logic flow during conceptual design.

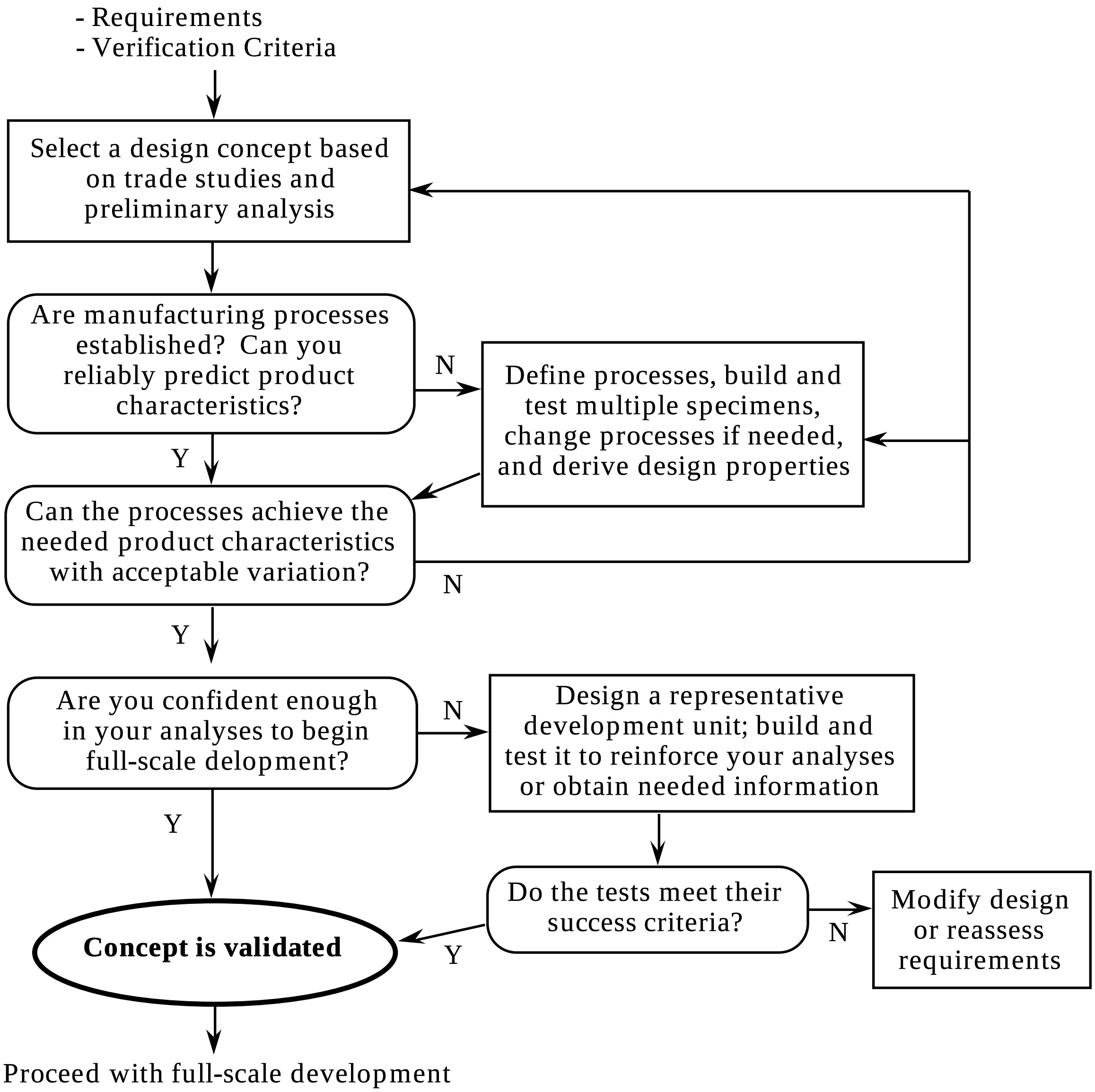
- Requirements
- Verification Criteria
Select a design concept based on trade studies and preliminary analysis
Are manufacturing processes established? Can you reliably predict product characteristics?
N
Define processes, build and test multiple specimens, change processes if needed, and derive design properties
Y
Can the processes achieve the needed product characteristics with acceptable variation?
N
Y
Are you confident enough in your analyses to begin full-scale delopment?
N
Design a representative development unit; build and test it to reinforce your analyses or obtain needed information
Y
Do the tests meet their success criteria?
N
Modify design or reassess requirements
Concept is validated
Y
Proceed with full-scale development

Fig. 15-1. Verification Logic Flow during Conceptual Design. (Adapted from SSAM[1], Fig. 11.2.)

The design phase of full-scale development culminates with the Critical Design Review (CDR), for which the goal is to agree, contractor and customer, upon the design and commit to production. To obtain such a commitment, the contractor's engineers must instill confidence. In other words, they must, to the best of their ability, verify compliance with the requirements by analysis, supported by any additional, needed development testing.

When verifying a requirement by analysis, it's particularly important to have well thought-out criteria for that analysis. Examples include thermal margins, weight-growth allowances, and structural factors of safety. Such criteria are intended to ensure analysis quality given uncertainty and the human element and also to ensure a high probability of

[1] *Spacecraft Structures and Mechanism: From Concept to Launch*, 1995, Sarafin, ed., Microcosm and Kluwer, publishers.

success given random variables. As with all aspects of verification, establishing sound criteria is the responsibility of the contractor, but the contractor must remember that the criteria have to be acceptable to the customer as well.

We use these criteria to judge the acceptability of analysis as our basis for committing to production, but the decision is easy only if the criteria are met. In this event, unless we've uncovered problems not encompassed by the criteria, the decision is a no-brainer. If the criteria are not met, we still might decide to accept the design and build the product. Recall that criteria differ from requirements in that they relate to risk. Failure to meet our criteria simply means there is more risk than we originally intended to accept. But other things have changed, too. We've invested time and money in developing this design, and it would be costly to modify it or start over. Before accepting this consequence, it makes sense to assess the risk implied by failing to meet the criteria.

This, you see, is the key to effective use of criteria: Don't get emotionally attached to them—and don't let them become a "black box" that no one truly understands. If a criterion is not met (e.g., negative structural margin of safety), strip away the jargon and strive to understand the risk.

Figure 15-2 portrays the verification logic flow during full-scale development. The conclusion that justifies proceeding to production—that the design meets requirements—is a preliminary conclusion that we'll revisit later in the program.

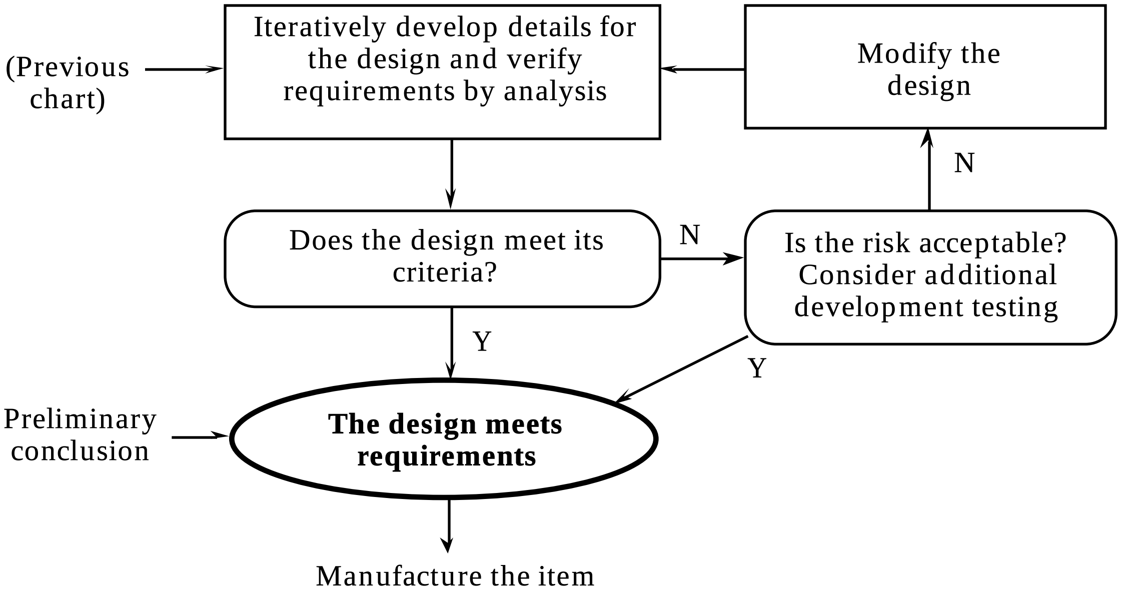
(Previous chart)
Iteratively develop details for the design and verify requirements by analysis
Modify the design
N
Does the design meet its criteria?
N
Is the risk acceptable? Consider additional development testing
Y
Y
Preliminary conclusion
The design meets requirements
Manufacture the item

Fig. 15-2. Verification Logic Flow during Full-Scale Development. (Adapted from SSAM Fig. 11.3)

In manufacturing, the objective is to build something that conforms completely to the specified design. Because processes, including tools and people, aren't perfect, things can go wrong. The first decision regarding verification of manufacturing quality is how well we'll be able to control the process, or how much time and money we're willing to spend learning to control the process. Remember, any organization can say their processes are controlled, but full control to the point where product inspections and tests are unnecessary requires meaningful, supporting statistics from data collected by inspecting or testing the products of those processes. I compared the philosophies of selective inspection and 100% inspection in my last article, so here I'll address the outcome of inspection.

An inspection is much the same as an analysis or a test, complete with its own pass/fail criteria. Just as we don't discard a design based solely on failure to meet the criteria, we don't necessarily scrap hardware that fails inspection. Again, failure to meet the criteria simply means more work is required. We write a discrepancy report to document the problem and then get the right people to assess it. Sometimes the analysis will still satisfy its own criteria, even though the product did not satisfy certain features within specified tolerances. Other times, as a result of the discrepancy, we can no longer satisfy the design criteria. If so, we still might accept the product, depending on our assessment of risk. The two main outcomes of the discrepancy report and the subsequent investigation are (1) what we will do with the discrepant item and (2) what we will do to ensure the problem will not occur for subsequent builds. The second outcome requires that we first find the cause of the problem. Figure 15-3 shows the flow of verification logic during manufacturing.

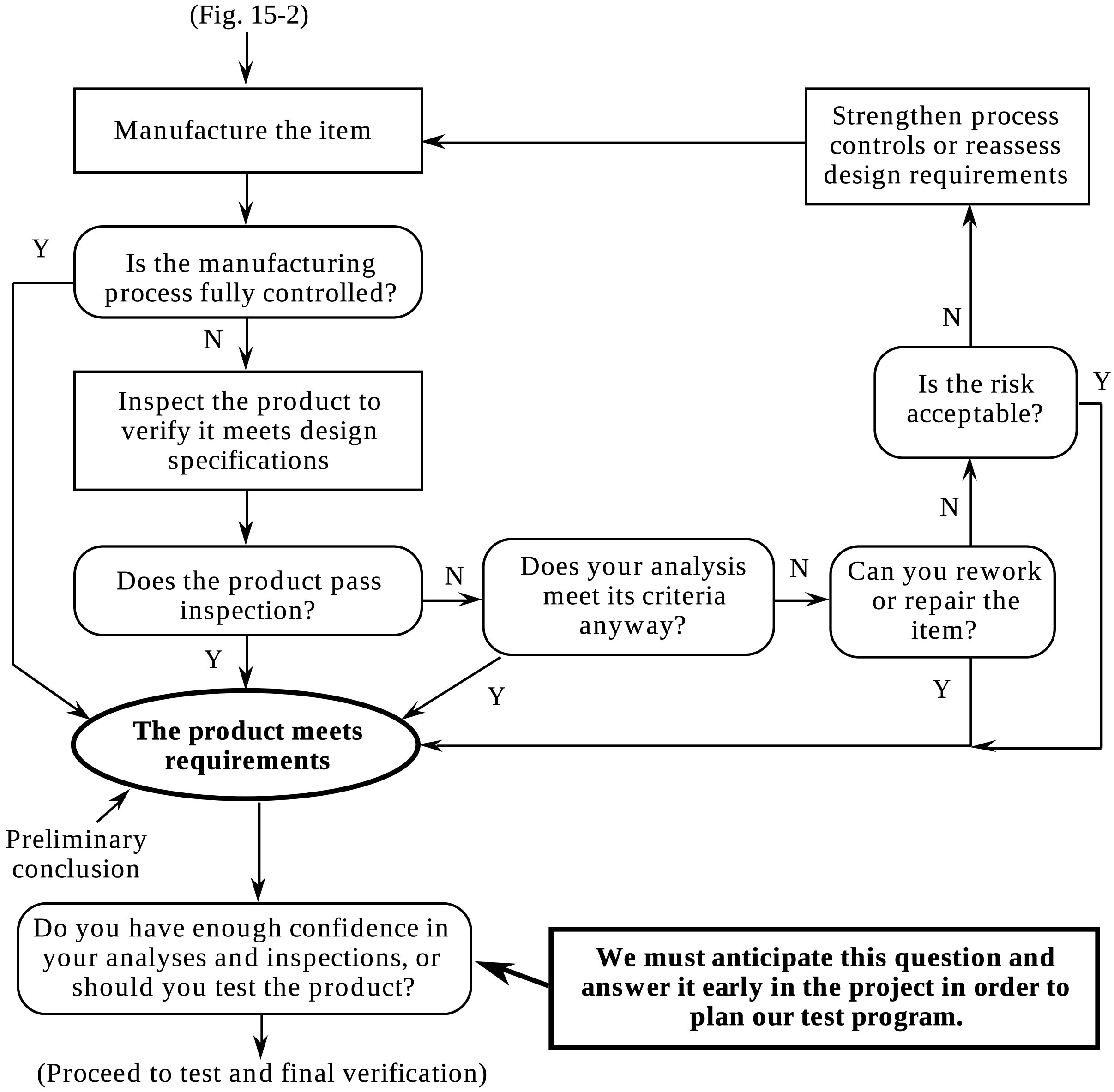
(Fig. 15-2)
Manufacture the item
Strengthen process controls or reassess design requirements
Is the manufacturing process fully controlled?
Y
N
Inspect the product to verify it meets design specifications
Is the risk acceptable?
N
Y
Does the product pass inspection?
N
Does your analysis meet its criteria anyway?
N
Can you rework or repair the item?
N
Y
Y
Y
The product meets requirements
Preliminary conclusion
Do you have enough confidence in your analyses and inspections, or should you test the product?
We must anticipate this question and answer it early in the project in order to plan our test program.
(Proceed to test and final verification)

Fig. 15-3. Verification Logic Flow during Manufacturing. (Adapted from SSAM Fig. 11.4)

Note the conclusion we are trying to reach after manufacturing: The product meets requirements. As noted in the flow chart, this is a preliminary conclusion, as was our decision going into manufacturing that the design meets requirements. Nagging doubts may persist. If, for some reason, we are not confident enough to rely solely on our analyses and inspections, we test the product.

Of course, we can't wait until this point in the program to decide whether to test the product. We must project ahead when planning for verification in order to ensure adequate time and budget. We go through the logic I've been describing, based on our design concept, the planned method of construction, the experience of our staff, our organization's experience with this type of design, and the current or expected state of process controls.

As shown in Fig. 15-4, there are three possible reasons for testing an end-item product:

– The analysis is in doubt

– The design is in doubt

– The process controls are in doubt

When we don't trust the analysis, we do an analysis-validation test, which is normally a nondestructive test intended to provide information that will make our analysis more dependable. Often we are trying to correlate a mathematical model with as-tested behavior of the product. For example, in a modal survey test, we try to individually excite a structure's key modes of vibration and determine natural frequencies, mode shapes, and damping. We use this information to improve the accuracy of our finite-element model, which will then be combined with models representing the rest of the launch-vehicle / payload system to predict launch loads.[2]

If the analysis is in doubt, the design based on that analysis is usually in doubt also. In this case, we ideally would do a qualification test on a dedicated, nonflight unit. As noted in my previous article, the purpose of a qualification test is to demonstrate the adequacy of a design.

When the process controls are in doubt, we test each product for acceptance. We often refer to an acceptance test as a "workmanship" test, but let's stop and consider the message this terminology sends to the workers, or technicians. We essentially are saying we don't trust them to do a good job. W. Edwards Deming said that, if quality is poor, it's not the fault of the workers, it's the fault of the system or process. In other words, something outside the control of the worker, such as poor-quality raw materials or tools, is usually at fault. Workmanship is only part of the process, and we do acceptance testing when we don't fully trust the process.

It's quite likely that, for any product, we'll feel the need to do all three types of tests, analysis-validation, qualification, and acceptance. For each test, note the role of criteria shown in Fig. 15-4. In all cases, failure to meet the criteria means we must assess risk.

[2] This analysis is called *coupled loads analysis* because it's done with combined (coupled) models. The full process of developing and combining models, calculating loads, and assessing loads is called a loads cycle. The typical payload is hard-mounted to the vibrating launch vehicle, unlike how a truck's payload is normally isolated from vibration with springs and shocks, so coupled loads analysis to support design is essential for sound engineering. The final loads cycle, using test-verified models, is called the *verification loads cycle*. The test-verified structural model will also be used for final verification of the vehicle's control system, which can be influenced by vibration.

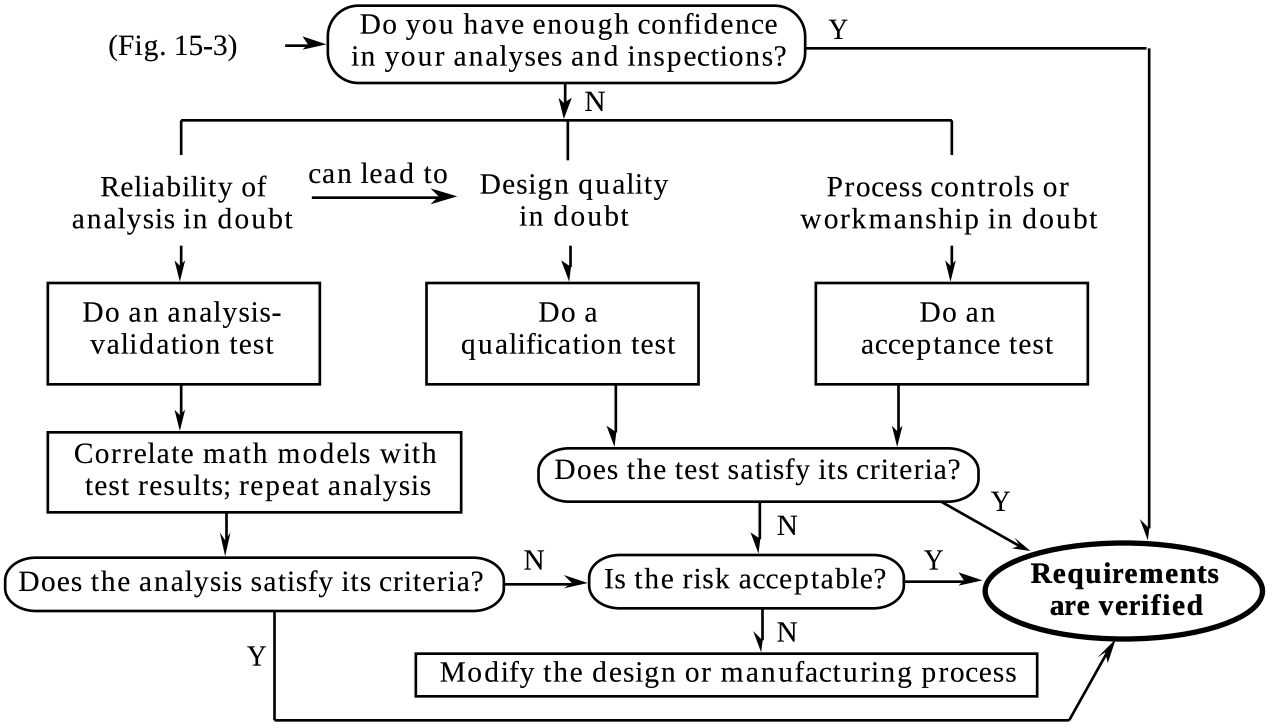
(Fig. 15-3)
Do you have enough confidence in your analyses and inspections?
Y
N
Reliability of analysis in doubt
can lead to
Design quality in doubt
Process controls or workmanship in doubt
Do an analysis-validation test
Do a qualification test
Do an acceptance test
Correlate math models with test results; repeat analysis
Does the test satisfy its criteria?
Y
N
Does the analysis satisfy its criteria?
N
Is the risk acceptable?
Y
Requirements are verified
N
Y
Modify the design or manufacturing process

Fig. 15-4. Logic Flow during Final Verification. (Adapted from SSAM Fig. 11.5)

So, when budgets are getting squeezed, how do we go about reducing the cost of a test program? In my opinion, as I've stated throughout these articles, it's a mistake to desert a sound engineering process, such as that shown in the above logic flows, in order to reduce planned cost. If testing is too expensive, we invest up front in development testing and in finding a simpler design in which we can be confident with minimal testing.

Planning to Avoid the Need for Acceptance Testing

Increased build quantity is a situation that is becoming more common as more uses for satellite constellations are conceived. If, say, we are planning to build 100 spacecraft of identical design, of course we would like not to have to test all of them. In a recent course I taught, I posed this problem to the class of practicing engineers, which I divided into teams. I asked them to develop a plan for verifying that the spacecraft could withstand mission environments. The resulting discussion went something like this:

"We'll test only the first five spacecraft, and we'll control our manufacturing processes."

"How will you control your processes?" I asked.

(Pause) "With written procedures, controlled temperature and humidity, inspections—all the usual stuff."

"How will you know whether your processes are controlled well enough?"

(Silence and a thoughtful look.)

I continued: "What happens if environmental testing finds problems in one or more of the first five?"

(Pause) "I guess we'll have to test the others, then."

"But you can't. You planned to test only the first five; you don't have time or money to test the rest. It's too late!"

I use this example to drive home the importance of extensive process development, supported by statistical data collected from development testing. Even if the answer to the above problem is to build, say, an extra five spacecraft to make the constellation redundant, the same issue exists. Only with meaningful data can you decide whether your processes are controlled well enough to support your plan. Deciding to test only the first few, even with a redundant system, without supporting data is not sound engineering. It's irresponsible.

Similarly, any test that you are considering deleting can be done so responsibly only through a sound understanding of its purpose and how it fits into the above logic flows. Said another way, if you don't understand why a test is normally done, you can't responsibly decide to do away with it.

Having the Right Attitude about Verification

As I discussed in the sixth article of this series (Principle #1), the most important thing in a space program is commitment to success. Having the right attitude also means ensuring your customer is confident.

I like to keep things simple (Principle #7), and that applies to verification, as well. Here's a two-step process that I believe captures the essence of verification and quality assurance:

1. Make sure your product will do what it's supposed to do.
2. Convince your customer that you have done so.

Most of this article and the previous one have addressed the first step. The second is nearly as important. It starts with the right attitude, and it requires communication skills. You might be the smartest and most diligent engineer in the world and have become completely confident in your product. But, if you can't—or don't take the time to—instill that confidence in your management and your customer, it won't mean much. Without confidence, the people who have the most at risk will spend money—gobs of it. Often, because those people don't know as much as you about your product, that money will not be spent wisely. Sometimes the wrong things will be done, things that actually jeopardize success. If you don't buy the argument that your customer deserves to feel confident, at least recognize that, by not making the effort, you are potentially driving risk as well as cost.

Recognize also that having the right attitude regarding verification does not mean striving for zero chance of failure. While 100% success is a wonderful underlying goal, the facts tell us that aiming for 100% probability of success is not cost-effective. As Fig. 15-5 indicates, each dollar spent reducing risk is progressively less effective. Because of random variables and uncertainty, we can never make failure impossible, but we can certainly go broke trying to do so. Our goal is to make our plans, criteria, and standards consistent with a cost-effective probability of success.

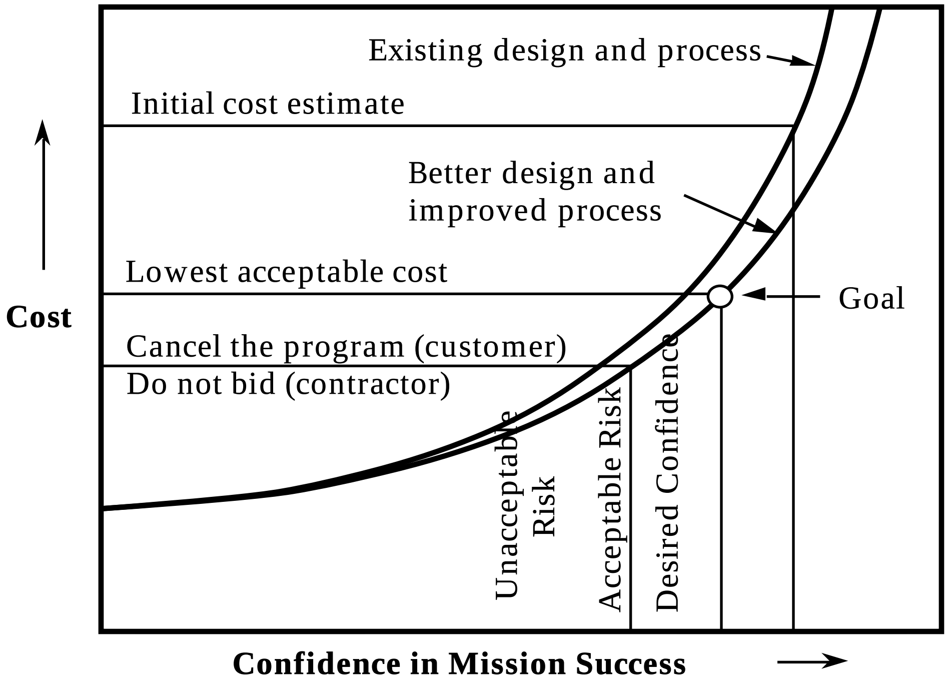

Fig. 15-5. Cost Versus Confidence. The best way to reduce cost is to lower the curve. We might also be willing to accept more risk, but we must understand the risk well to make sure it’s acceptable. (Adapted from SSAM Fig. 11.1)

There are three main ways to reduce cost of a program:

1. Improve the design
2. Improve the process (including personal effectiveness and management)
3. Accept more risk

While our emphasis should be on the first two, there’s nothing wrong with accepting risk to save money, as long as the risk is understood to be low. Accepting risk, though, should never be our main strategy. Unfortunately, we are tempted to do so because the first two ways to reduce cost require investment. As Fig. 15-5 points out, if, after improving the process and the design, we don’t have enough money to do a space mission with no more risk than is acceptable, we should cancel the program. It’s the only responsible thing to do.

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.