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 #16

Follow a Sound Engineering Approach
Part 5: The Engineer's Responsibility

November, 2001

Tom Sarafin

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

The preceding four articles in this series dealt mostly with a sound approach to systems engineering. This one is aimed at engineers contributing to system development in more specialized roles.

As I collect my thoughts in preparing to write this article, two experiences come to mind, both involving college engineering students. The first was when I myself was a senior at Ohio State, in the spring of 1979. I was waiting after class to ask a question I no longer remember, and I overheard the discussion my professor had with two class members in line ahead of me. We had just received scores on a homework problem, and the two students, who had clearly worked the problem together, were arguing their case. The assignment had been an engineering design problem that required iteration: make an assumption, work through a page or two of calculations, check to see how closely the result agreed with the initial assumption, and then revise that assumption accordingly and work through the problem again. In the two prior years of engineering courses, we had worked many problems such as this, in which we were expected to iterate until the answer had sufficient accuracy. In other words, we were expected to follow a sound engineering process.

In this problem, our professor had helped us reduce the needed iterations by suggesting a value for the initial assumption. As I recall, the discussion with the two students went something like this:

"Why did you mark our papers wrong?"

"Because you got the wrong answer. It doesn't agree with the initial assumption."

"But you told us to make that assumption!"

"Yes, but you didn't check the assumption!"

The two students looked at each other for a moment with eyes widened. Then one said to the professor with genuine outrage, "You didn't tell us we had to check the assumption!"

As I listened, I'm sure my jaw gradually dropped. We were six weeks from graduation, from starting our careers as engineers—and these two still had not grasped the concept of iteration.

The second story I want to share is more recent. I was helping a senior in mechanical engineering with a class project of designing a small structural assembly. His immediate problem was to estimate the bending stress in a machined plate, which was the critical part of the structure, and conclude whether the structure was safe. I explained how the plate was acting like a beam and showed how the loads were introduced and reacted. I told him to calculate the section properties, which he had been taught how to do. He would then use beam theory, which he had also learned, to predict the peak bending stress and compare it to the allowable stress for the material.

The next week, the student I had helped presented to the rest of the design team that the plate was indeed safe, with a peak bending stress of only 2 ksi (2000 pounds per square inch). Already knowing the answer was about 40 ksi, I asked him to show me his analysis during a break. He showed me a page full of sloppy calculations, written haphazardly and barely legibly. I pointed out that his calculated section moment of inertia was about two orders of magnitude too high. I showed him how a check of a simple rectangular section would have found his error. I chastised him for sloppy work and told him to repeat the analysis in a manner neat enough to be checked.

After he had revised his analysis, he asked me to review it. He had now predicted 220 ksi—wrong again. This time, he had properly calculated the moment of inertia, but the bending moment he had calculated was about five times too high. What he did wrong was not apparent because he had not shown any calculations. Once again, by not using a disciplined process—which, in this case, would have included drawing a moment diagram to identify the peak moment—he had arrived at a wrong answer and was unable to find the error himself.

These examples are of students, not practicing engineers, but similar stories abound in the engineering ranks. Unfortunately, the ability and discipline to follow a sound process do not come automatically with an engineering degree. We cannot take them for granted, even though they are perhaps the essence of engineering. Without a sound approach, not only are mistakes made, but time and money are wasted as well.

A generic process that can be used effectively for almost any engineering activity is quite similar to the one defined in Article #12 of this series for developing a system:

1. Define objectives
2. Identify requirements and constraints
3. Plan the task
4. Assemble information and tools
5. Make assumptions, as necessary
6. Perform the task
7. Confirm assumptions and iterate, as needed
8. Check results
9. Document the activity throughout the process
10. Implement the results

1. Defining Objectives

Every engineering activity should start with clear objectives. This seems obvious, but how often do you take the time before starting a task to write down your objectives? And how often, after the job is finished, do you wish you had done so? Often it's hard to put objectives down in words, but making an effort in this regard will help prevent you from wasting time and money on things that provide little true benefit. If you can't define the objectives for your assignment, you need to talk again with your boss.

Team members working to different objectives is a common problem. I once served as lead stress analyst for a large spacecraft program. We received news from the launch-vehicle organization that recent flight data showed the acoustic environment was higher than predicted. Our vehicle was being assembled and integrated in preparation for acoustic testing. Of course, program management and our customer wanted to know if the more severe environment was of any concern to our spacecraft. My stress group was ready to assess the solar panels, mounting brackets, and other structures that were sensitive to acoustic loading and the induced random vibration. (Other folks were to assess the electrical components for random vibration.) First, though, we had to wait for Jerry in the dynamics group to predict vibroacoustic responses.

Each day, someone—a manager, a customer representative, or a structural design engineer—would ask me if the vehicle could handle the new test environment, and each day I said we still didn't have loads. Every few days, I would bug Jerry: "Are you done with your analysis yet?" The answer was always the same: "Not yet."

Finally, with the test scheduled to start in a couple weeks, I confronted him: "Jerry, when will your analysis be done?"

He looked at me in surprise and said, "Oh, probably the night before the test."

With a sinking feeling, I asked, "Jerry, exactly why are you doing this analysis?"

He answered without hesitation: "So we have predictions to compare the test data with."

"No!" I said. "You're doing this analysis so we can assess whether the vehicle can withstand the environment!"

A lack of communication? Obviously. This example shows that, even when you define your objectives, your task can have little value if they're not consistent with those of the program.

2. Identifying Requirements and Constraints

Without a doubt, you'll have to meet certain requirements when performing your task. Objectives usually must be tempered by constraints on time and budget. Who will use the results, and when do they need them? It's also likely that standards or acceptance criteria will apply. The "customers" for your task, the people who will use the results, may also drive requirements pertaining to the product (e.g, content or format).

From the objectives and needs of your customers, identify the required accuracy. Engineers tend to fall in love with accuracy, whether that accuracy is actual or perceived. When constrained by time or budget, you can usually simplify your approach at the cost of accuracy. A trait of a truly effective engineer is the ability to identify the accuracy that is cost effective. A companion trait is knowledge of the simplest way to obtain that accuracy.

3. Planning the Task

Planning entails devising or finding an approach that will meet the objectives and requirements. It includes identifying inputs (what you need) and where you will obtain them, and outputs (what you will provide) and who you will provide them to. Lack of planning will become apparent, most likely by missing a deadline (when someone needs the results of your activity).

Planning is often iterative, especially for complex tasks, such as design or testing. You start with a preliminary plan, do some work, unearth information, and revise the plan.

Depending on the process you design, you may want to test it before using it for the final activity. As with most aspects of engineering, common sense applies: If the results of using the process are permanent, it makes sense to try it first on something that has lower consequence of failure. Manufacturing processes fall into this category, as do many test processes.

4. Assembling Information and Tools

Your plan should tell you what you need here. Throughout your career, cultivate sources of information and a network of experts who can provide what you need or lead you to the right source.

5. Making Assumptions

Assumptions are made in engineering activities for two main reasons. Sometimes, as in the first example related in this article, you need an assumption as a starting point for an iterative engineering process. Probably more often, you make an assumption to fill in for information that is not available. In either case, it's important to document your assumptions and make sure you revisit them later. (See Step 7, below.)

6. Performing the Task

This, of course, is what you wanted to do from the start! It takes discipline to keep from jumping into the task too soon. The above steps don't have to be time-consuming to be effective.

7. Confirming Assumptions

The assumptions you made were for the purpose of getting you this far in the process. Now that you have more information, remember to go back and check those assumptions and iterate, if needed. Often, your task by itself will not give you the information you need to confirm your assumptions. In such cases, especially when inaccuracy or error would be damaging, it makes sense to assess the sensitivity of your assumptions. In other words, vary your assumptions within reasonable ranges and see what happens to your conclusions.

As an example, consider development of a finite-element model to mathematically represent a spacecraft structure. Such a model will be used to predict launch loads, ensure the spacecraft meets certain constraints imposed by the launch vehicle, and verify that the spacecraft's control system will not be confused by vibration. The model's accuracy will depend on your assumptions. The model will be wrong, to some extent. All mathematical models are. Even if your program plans to test the structure to confirm the model, that test will happen too late to avoid a high cost impact of model inaccuracy. If your model is wrong, the structure will have been built to the wrong loads, and the spacecraft's control system will have been designed to be compatible with the wrong modes of vibration. Your customer may not have required that you assess how sensitive the analysis results are to your modeling assumptions, before the program commits to those results, but doing so only makes sense. It's your responsibility to confirm such influential assumptions.

Failing to confirm key assumptions because you lack the time to do so is inexcusable. Everyone is pressed for time; it's the nature of our business. Part of your job is to plan out an approach or level of detail that leaves enough time to confirm your assumptions and check the results.

8. Checking Results

Checking the results of an engineering activity can be a passive, after-the-fact process, but it's far better to do most of the work up front. The best engineers make it a habit to estimate answers ahead of time, and they never embark on a black-box process, such as those done by purchased software packages, without knowing what they expect to find.

The same philosophy applies to testing. Before doing a test, work out what you think will happen along with several other possible outcomes. Then plan out what you would do in each case.

This practice serves several purposes when applied to a test: It builds understanding, it shortens the test (and time monopolizing a standing army of test personnel), and it enables you to recognize bad test data. Data acquisition during tests is not flawless; many aspects of the process can break down. Without good predictions backed by confidence, which is in turn based on a sound understanding of the problem, you will unknowingly accept bad data, and many wrong decisions may be made as a result.

9. Documenting

Documentation has gotten a bum rap recently as we've tried to find ways to reduce cost. Attacks on documentation are understandable because many generated mounds of paperwork go untouched in space programs. But well-done documentation records key things that you otherwise would forget, and it allows others to understand what you've done. Unless you are the only one on the project and have a perfect memory, much of your effort will be wasted if you don't document it well.

A common trap is to leave documentation for the end of the task, scribbling notes along the way only to the extent needed to allow you to proceed to the next step. Experienced engineers know it doesn't work this way. Leaving documentation for the end will ensure an inadequate record and usually will lead to undetected errors. Remember the second story I related in beginning this article. If you haven't yet been on the guilty end of a similar experience, you probably will be. Your documentation throughout the task should be complete and neat enough to capture key information and enable self checks.

Traditionally, after the activity was completed, daily records have been painstakingly transcribed into a neat final report. Today's computer tools make the process much more efficient. By planning ahead and working within the framework of a good outline or template, your notes, sketches, and rough analyses can, with minor editing, become parts of the final report

The format and contents of your documentation should reflect the engineering process. The things I've discussed above as being important to the process are the things you should document. Objectives, requirements, methods, assumptions, checks, and results—all should be laid out in a logical format. Remember: A good technical document makes it easy for people to find what they're looking for. Remember also: The person trying most often to find things in your document will most likely be you.

10. Implementing the Results

After following such a sound engineering approach, you can confidently and proudly offer up the results of your activity for use. Go home and have a beer (or other beverage of choice), and reflect on how fortunate your organization is to have you on the team!

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.