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.

Article #5

Ten Principles for Doing Things Right

May 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

My previous two articles dealt with recurring problems in the space industry, with the objective of building an understanding of the overall problem before we try to solve it. In this article, I'll try to draw some conclusions and start to explore solutions. I admit up front that I don't have all the answers, I don't know all the right paths to take. But I can try to provide a compass.

From studying these problems, first off, it would seem there are many things we, as an industry and as individuals, don't understand very well regarding not only the unique challenges of our industry but also basic engineering and project management. We appear to have problems with communication, cooperation, and morale. And, apparently, we don't consistently consider downstream activities during design, do adequate testing, take care of details throughout product development, and generally take responsibility for our products and tasks.

We can't solve such problems by cutting budgets and putting more pressure on our people, and we can't solve them with more powerful computers and software. Instead, we must get to the root: Why is the quality of engineering not where we want it to be? Why is it that we don't attend enough to downstream activities and details, and why is there a lack of personal responsibility? Low morale could explain some of these things, at least in part, but why is morale low? Why is it that people and organizations don't cooperate more? If we look closer, we begin to uncover some root problems:

- **A short-term focus**
  - Lack of investment in developing people, technologies, and methods
  - Structuring programs to react to problems rather than avoid them

- **Compartmentalizing people into ever narrowing, specialized fields**
  - Leads to tunnel vision, competition, and lack of insight and trust
    - which in turn leads to poor communication, cooperation, and engineering
- **Outsourcing manufacturing and test; closing down in-house facilities**
  - Fewer chances for design engineers and analysts to learn from actual hardware
  - Development testing is becoming more expensive and is thus more rarely done
- **Practices that demoralize people and take away responsibility**
  - Providing inadequate time for good engineering, with the burden on engineers
  - Specifying how products must be engineered and verified
- **Emphasis on cost rather than quality, mission success, and a satisfied customer**
  - This, in turn, stems from a procurement system that awards contracts based on proposed cost more than anticipated or demonstrated quality
- **A push by customers to do more mission for the money without first working with contractors to solve the above problems**
- **A difficulty in attracting and keeping talented engineers, stemming from the above**

To solve our problems in the space industry, which we desperately need to do, we must address the above issues. We must be willing to invest in the future and find ways to more effectively educate our people, and we must root out and eliminate anything that impedes teamwork. If we are successful here, we can begin to trust our people, so we can give them more responsibility and time to do their jobs, and otherwise treat them with respect. We also must dissect our process and fix it, or, perhaps better yet, sweep it off the table completely and reinvent it so that everyone understands it and knows how he or she contributes to it.

You think that's not a problem at your organization? That everyone understands how he or she fits in? At one course I taught, I had ten stress analysts from the same company silently staring at each other for ten minutes, stymied by my request to identify how they contributed to their programs, or, in other words, why their program manager wanted them onboard. If everyone on your program truly understands his or her responsibility and contributions to the whole—and anyone who understands the whole thing—count your blessings.

Finally, management—from the top down, customers as well as contractors—must fully embrace the notion that quality products and successful space missions are more important than cost savings.[1] I'm not saying cost isn't important; everyone must be conscious of it, but we must strive for total cost reduction by avoiding problems through quality rather than short-term savings that introduce unknown risks.

I can't overemphasize this point. Our people are the key to success, and, above all else, they must be committed to it. Space missions are just too complex, there are too many things that could go wrong for us to count on procedures and reviews. Everyone working on the program must be committed to success.

Nothing destroys that commitment in the troops more than the sense that management doesn't have it. When industry leaders and top management send the message down that we all have to accept more risk in order to reduce cost, the wheels

[1] The only exception here is when we are exploring new technologies, in programs for which the mission is R&D. In such cases, we accept risks of the unknown, but even then we should not compromise our quality system for the things we know how to do.

begin to come off the track. Budget is cut and we must now leave a few stones uncovered. Everyone becomes confused by a philosophy that goes against the grain of what they were taught that good engineering and quality work were all about. And if, at this point, they get the conflicting message that success is still the most important thing, they become cynical and wish they were somewhere else. The worst thing management can do is post slogans such as "Failure is not an option" or "100% successful missions" and then arbitrarily cut the engineering budget or decide without sound technical rationale not to do a costly test that the engineers believe is needed.

Do you recall a man named W. Edwards Deming? He, more than anyone else, taught quality to the Japanese. About ten years ago, Total Quality Management (TQM) became the management fad in the space industry. TQM mostly derives from Deming's teachings and his fourteen points, one of which was to eliminate banners and slogans. Most of us in the space industry didn't take the time to understand what Deming was saying, which was that banners and slogans demoralize the workforce. Sure enough, many people in the space industry were indeed demoralized by banners at their companies proclaiming "XYZ Company Embraces TQM" (which is itself quite ironic). I know people were demoralized because I was one of them, and most of the engineers I worked with felt the same way. Why? Because we knew that's where it would end. Nothing important would change. Deming said that if you truly fix the problems at your company, your people will hang up their own banners.

As it turned out, TQM didn't work in our industry. Deming's own experience was with high-volume industries, so he emphasized statistical methods of process control that aren't feasible when applied directly to low-volume industries such as ours. We couldn't figure out how to apply what he taught, so we were on to the next fad. The truth is, Deming's most important message had to do with principles that apply to all businesses.

Let's take a look for a moment at a utopian world. It's important to have a vision of where we're headed, even if we don't quite know how to get there. Constantly striving for that vision is the important thing. Here, in just a few words, is the key to solving our problems, the key to cost-effective, successful space missions:

Broadly and deeply knowledgeable, motivated people working together in a process they understand and believe in

- *Broadly and deeply knowledgeable*
  - Experience working in or with multiple disciplines; not overly specialized
  - Strong understanding of how their decisions affect the overall program
  - Strong understanding not only of theory but also of how things really behave, which can only be gained from experience: leaving their computers frequently and getting their hands dirty
- *Motivated*
  - Given clear, challenging responsibilities
  - Empowered to fulfill their responsibilities
  - Having influence
- *Working together*
  - Cooperation and trust between contractors and customers
  - Co-located, closely knit teams; no barriers
  - Having access to all needed information
  - All disciplines represented early

- *A process they understand*
  - Clearly defined objectives, responsibilities, and interfaces
- *A process they believe in*
  - Having a say in developing the process
  - Budget, schedule, and tools that are consistent with responsibilities and processes
  - All management decisions consistent with the notion that the most important thing is a successful mission

As Deming said, if quality is poor at your organization, it's not the fault of the workers (including the engineers); it's the fault of the system. And only management can change the system.

If we are to solve our problems, if we are to make this industry fun and thus attract the talented people we need, our leaders must lead us to visions that are based on sound principles rather than superficial fixes. I suggest here ten such principles.

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**

There is no great secret, here. These principles simply make sense, which is what makes them principles. Let's look to see what they are saying:

1. **Adopt the right attitude**
   - From top management on down, make quality and a successful mission the top priorities in all you say and do. Quality is everyone's job, not just that of the Quality Group.
   - Leave nothing to chance; address all issues.
   - Never cut budget or schedule arbitrarily to save money if it might jeopardize success.
   - If cost is a constraint that prevents you from doing things right, then reduce the scope of the mission. Be able to do it well, or don't do it at all.

- Remember, though, that "quality" also means a fair price, so strive for success as efficiently as possible.
- Acknowledge that no one deserves confidence throughout the program more than your customer does. Becoming confident in success yourself is not good enough.

2. Invest in knowledge and understanding

- Take time to understand the problem before attacking it.
- Much of the problem often has to do with variation. Strengthen your understanding of probability and statistics so you can recognize issues and more cost-effectively deal with them.
- Give people broad responsibilities and encourage broad involvement.
- Bring manufacturing and testing back in-house to make them more accessible to design engineers and analysts.
- Tactfully educate your customers and contractors, and allow them to educate you.
- Enable and reward experienced people for sharing their knowledge rather than riding them for all they're worth.
- Establish effective educational programs and encourage everyone to participate.
- Take responsibility for your own continuous education.

3. Instill ownership and responsibility

- Award contracts based more on quality—as assessed by the bidders' verification plans, quality systems, and history—than on proposed cost.
- Specify only what the product must be able to do (*requirements*), not how to demonstrate the product will do it (*verification*).
- Make verification the responsibility of the contractor developing the product, but make sure the contractor understands that verification includes making the customer confident (part of Principle #1).
- Clearly define responsibilities for everyone on the program, and then expect and allow everyone to fulfill them.
- Avoid conflicts of interest that reward cost savings.

4. Constantly seek ways to improve teamwork

- Make everyone part of the team: customers, contractors, technicians, secretaries—everyone.
- Hide nothing; build trust and enable teamwork by being completely honest with contractors, customers, and coworkers.
- Treat your contractors the same as you treat your customers.
- Co-locate team members who should work together most closely.
- Constantly improve your communication and people skills.
- Work with unions to get cooperation in the environment you are trying to create.

- Eliminate any competition between employees, and strive to make jobs more secure.

5. Follow a sound engineering approach

- Start with objectives rather than requirements.
- Develop and assess alternative concepts, considering the entire system.
- Be willing to accept performance penalties to enable a sound approach with available funds.
- Use iterative assessments to cost-effectively allocate requirements to system elements.
- Develop a verification plan based on true requirements—what the system or product must be able to do—and on key issues rather than on how the last program did it.
- Control manufacturing processes, thus build-to-build variation, when you can. Test each product when you can't.
- Ensure complete traceability between high-level requirements, low-level requirements, verification plans, and verification results.

6. Reduce total cost through good engineering

- Plan out the complete product life cycle early, and consider all events before releasing designs: manufacturing, test, vehicle integration, transportation, launch-site operations, mission operations.
- Design to control launch loads when you can, and design to tolerate uncertainty when you can't.
- Do development tests when you need more information to adequately predict a design's performance or capability.

7. Keep everything as simple as possible

- Recognize that complexity drives cost and adds risk, so make things simple even if it takes more time to do so:
  - Requirements
  - Designs
  - Analyses
  - Math models
  - Everything
- Automate repetitive tasks.

8. Establish an effective quality system that involves everyone

- Focus on preventing things that can go wrong rather than on getting certified for ISO 9000. Getting certified will be easy if you establish a truly effective quality system.
- Allocate broad responsibilities to functional disciplines, and then let them define their own standards and processes for ensuring those responsibilities will be met.
- Get everyone involved in order to develop understanding and ownership.
- Coordinate interfaces.

9. Be willing to accept risks, but only those you truly understand

- This is not at odds with Principle #1. Because of variables and unknowns, we can never make failure impossible, not with space missions and not with highway bridges. In all engineering fields, we've always had to balance risk with cost.
- Risk is defined by two things: probability and consequence of failure. Make sure you understand them both before accepting a risk. Taking a risk when you do not understand those two things is irresponsible.

10. Make sure everyone has enough time, resources, and freedom to do things right

- Constantly invest in making processes more efficient, but, until they are, don't cut budgets and compress schedules.
- Once we've successfully implemented the above principles, we'll find we can responsibly reduce the development budget and compress the time from concept to launch.

Note that we can't pick and choose from these principles; we have to accept all of them. We can't give our engineers more time up front until we've invested in their practical education. Our engineers can't take steps to avoid downstream problems if we don't give them enough time. We can't make our contractors responsible for verification if they haven't earned our trust and if they haven't accepted that part of verification is making the customer confident. If we don't have the right attitude, we'll never have an effective quality system. And on and on.

Beginning with next month's article, I'll individually address each of the above principles in more depth.

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.

Instar's Core Courses

- **DTR—Doing Things Right in Space Programs: A course for managers**
- **SDV—Doing Things Right in System Development and Verification**
- **USS—Understanding Spacecraft Systems**
- **SMS—Space-Mission Structures: From Concept to Launch**

Additional Instar Courses

- DASS—Design and Analysis of Space-Mission Structures
- USRV—Understanding Structural Requirements and Verification
- SPAD—Space Propulsion Analysis and Design
- OSPS—Overview of Space Propulsion Systems
- DAFJ—Design and Analysis of Fastened Joints
- APSIT—Avoiding Problems in Spacecraft Integration and Test
- GDT—Geometric Dimensioning and Tolerancing

Additional courses in work; customized versions available

For information on these courses, visit our website at instarengineering.com