Doing Things Right in Space Programs

This article is first in a series. 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 #1

How Can We Reduce Cost and Still Ensure a Successful Space Mission?

January 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

Over the past ten or so years, the biggest problem facing the space industry has been how to reduce cost. The problem isn't really that hard until we complicate things by insisting on having a successful mission. It's that qualifier that our industry has tended to overlook when planning a cost-reduction strategy.

Figure 1-1 shows how cost builds over a typical space program, which I've divided into three phases:

- requirements definition and design (*design phase*)
- production, integration, test, and launch
- operation

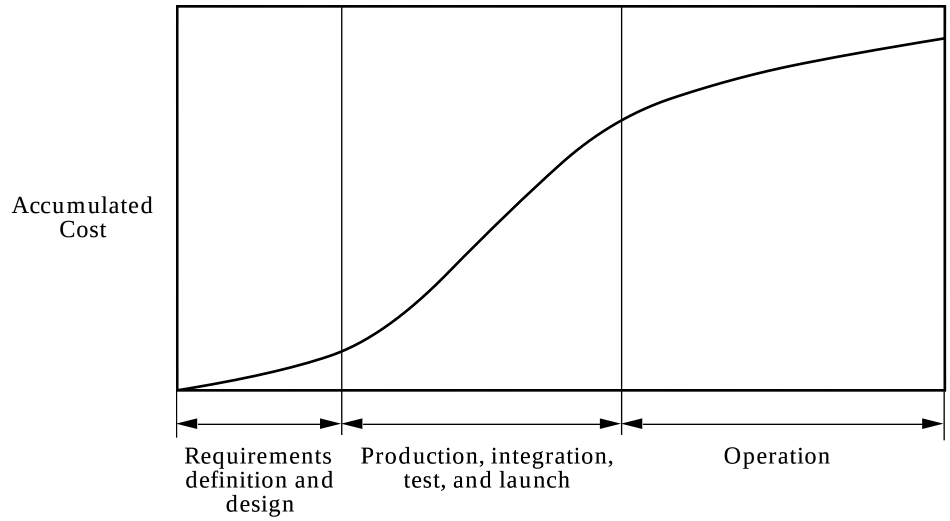

Fig. 1-1. Cost Profile for a Typical Space Program. Our challenge is to reduce the overall cost.

Let's take a moment to understand what Fig. 1-1 tells us. Notice how little of the total is spent in the design phase. A typical space program spends perhaps 5% to 10% of the total during this phase, which we'll say ends at the Critical Design Review (CDR).

Note, however, that the design is what determines most of the total cost. As Fig. 1-2 shows, once we release the design for production—the goal of CDR—our program has committed probably 90% to 95% of the total cost. We won't know how much the total cost will be yet, but there's very little we can do to change that total after we've released the design—other than choosing to accept increased risk, which I'll address in a later article. Obviously, the design phase is the most influential. It is here we must focus if we are to reduce cost without jeopardizing the mission.

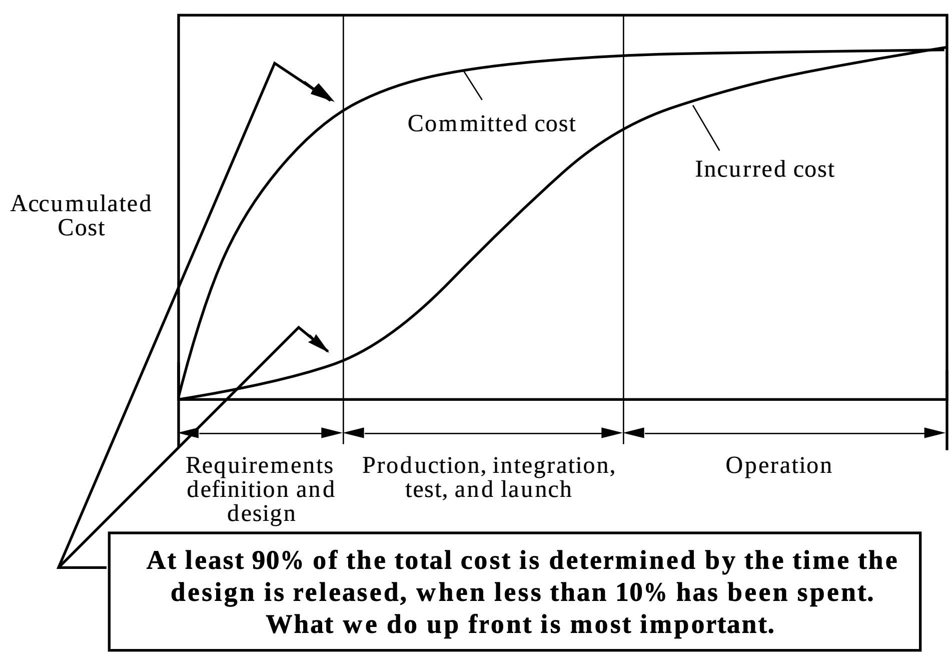

Fig. 1-2. Cost Profile Showing Committed Cost for a Typical Space Program. The ***committed*** cost is the amount locked in at any given time in the program.

Please note that I did not invent this way to present cost, and I'm not the first to recognize the message it contains. I remember seeing charts similar to those in Figs. 1 and 2 in an article (that I can no longer find or remember the author of) on Integrated Product Development in 1991. I thought, wow, what a simple yet powerful message! Somehow, though, the message has been lost.

One of the things I will harp about in this series of articles is the importance of understanding a problem before trying to solve it. We often think we understand a problem when all we see is a small bit of it. We can understand a clam to some extent by looking at its surface, but we reach a much greater understanding by opening it.

It's common knowledge that Albert Einstein was a poor student. He once said it was an advantage being retarded, unable to understand things that came so easily for everyone else. It was that very "disability" that made him keep trying to understand space and time, things the rest of us thought we understood all along.

So why do I bring this subject up now? Well, let's look at how we have commonly tried to solve the problem of cost in the space industry. The usual approach has been to cut the budget uniformly throughout all phases of the program. For example, if we want to reduce the cost of a space mission by 30%, we take the cost estimates from all the program phases and departments and cut them by 30%, as shown in Fig. 1-3, including the most influential phase: design. In other words, we try to force the engineers to do things faster, better, and cheaper by giving them less time and money.

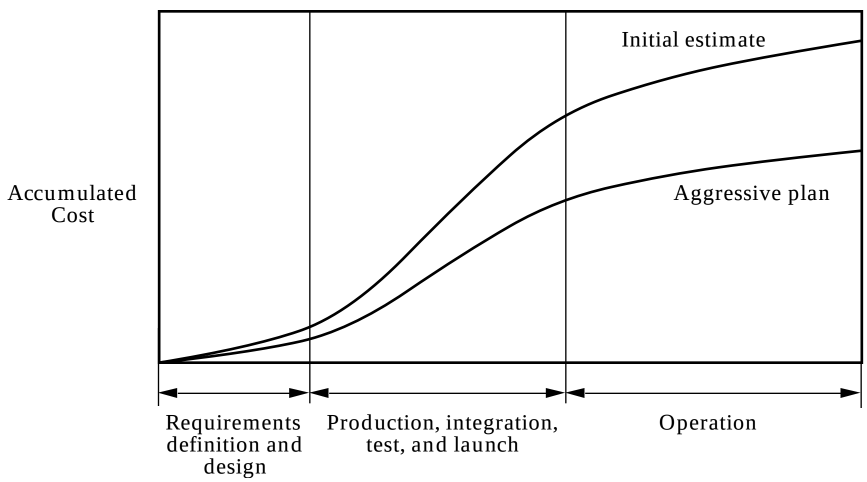

Fig. 1-3. A Common Plan to Reduce Cost. Cut the budget proportionally throughout all departments and phases of the program. This plan is bound to fail.

Doesn't sound like much of a plan, does it? But if you don't believe me, look around the space industry. Over the past seven years, I've taught over 70 short courses to more than 1500 space-program engineers and managers from dozens of organizations. Everywhere I have gone, the answer has been the same. The main plan for reducing cost has been to cut the budget uniformly throughout all phases of the program.

When this approach has actually reduced total cost, it has done so only by chance. The program has most likely taken risks when problems arose—or cut back so much they haven't even identified those problems—and gotten lucky … or not, as evidenced by the recent disturbing run of mission failures. (Table 1-2 summarizes recent launch failures.) Trying to reduce cost by arbitrarily cutting budgets has no hope of success if we define *success* as reducing cost without jeopardizing the mission. Figure 1-4 shows what usually happens with such a plan.

Table 1-1. Eight Launch Failures in Twelve Months. We could compile a similar list of spacecraft that didn't work after successful launches. An outcome of cost cutting? (Information compiled from press releases and articles in ***Aviation Week & Space Technology*** magazine.)

Date	LV and Payload	Problem Description	Cause (known or suspected)
8/98	Titan IVA/Centaur; military satellite	Guidance computer lost power; explosion	Unnoticed damage to electrical wiring
8/98	Delta 3; HS 601 HP	Control system went unstable; explosion	Improper assumptions in math models; poor communication
9/98	Zenit; Globalstar (12 satellites)	Second stage failed; all 12 satellites destroyed	Computer failure
4/99	Titan IVB/IUS; missile-warning sat	Upper stage did not separate; wrong orbit	Thermal wrap on electrical connector prevented separation
4/99	Athena II; Ikonos commercial sat	Payload fairing did not separate; payload in ocean	Electrical signal interrupted because of pyrotechnic shock
4/99	Titan IVB; Milstar	Upper stage misfired; Milstar in useless orbit	Data input error; recognized, but no one followed up on it
5/99	Delta 3; Orion 3	Engine failure, explosive rupture; useless orbit	Voids in a brazed joint; requirements poorly communicated
7/99	Proton; classified Russian comm. sat	Second-stage explosion	Weld failure in engine turbine cover

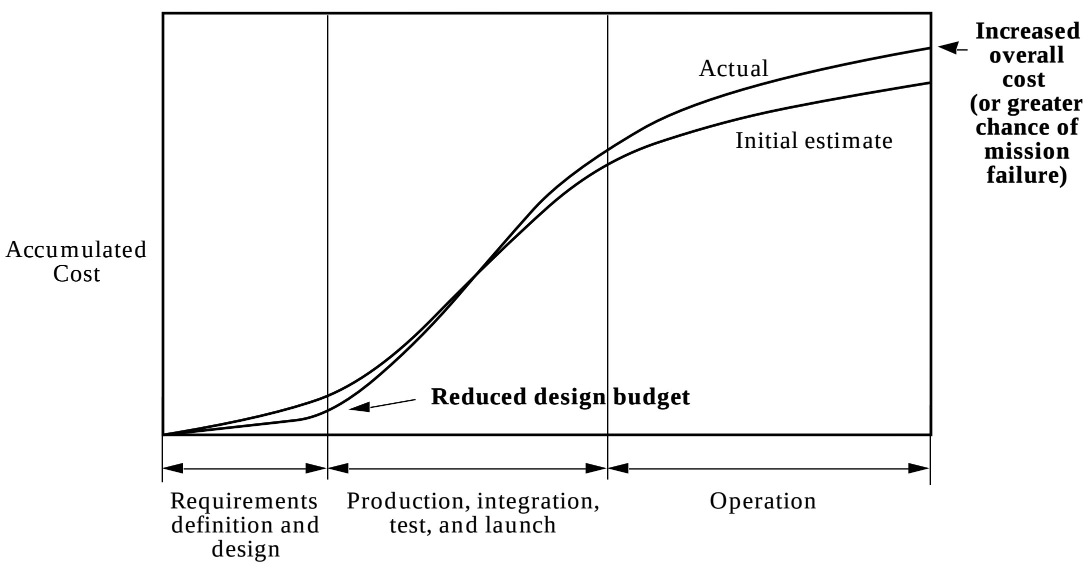

Fig. 1-4. What Usually Happens when We Cut the Design Budget. Problems arise after the design is released. We spend a lot of time and money building things, analyzing them, testing them, and putting out "fires." Or we fail to detect problems (or, worse, fail to properly address them) and have mission failure.

Why does the overall cost increase when we cut the design budget? Let's open the clam and try to understand it.

For a moment, put yourself in the shoes of a design engineer. (Perhaps you are a design engineer.) You join a program, get your assignment, and form a rough estimate of how much time it will take you. Let's say ten months. Your boss, however, gives you only five months, despite your protests. You wait for your boss to teach you how to do your job more efficiently, give you software that will automate many of your tasks, or otherwise improve the system. But no improvements show up—and you certainly don't have time to improve the system yourself.

What do you do?

You'll prioritize if you're like most engineers. You'll break your tasks into those that must get done and those that should get done; then you'll emphasize the former and only address the latter if you find the time, which you won't.

For example, you must release your design for production in five months, so you find a way to do that. But, along the way, you find no time to talk to the people who have to build the product, so you don't get their inputs to make it easier to manufacture. You aren't able to get with the test group to find ways to make your design easier to test. You should, but you just don't have enough time.

This happens so often: one person who is under the gun to get a job done makes an assumption that simplifies his or her job but passes the problem on to the next person in line. Sometimes the problem gets amplified by an order of magnitude or more.

A case in point is problems with launch loads, which besiege so many spacecraft programs. On one program I supported as a consultant, predicted loads in the spacecraft primary structure went up by a factor of three after everything was built and tested. The source of these new loads was a loads analysis that used, for the first time, test-verified finite-element models of the spacecraft and launch vehicle. (We refer to such analysis as the *verification loads cycle*.) Our spacecraft could not withstand the new loads. Our options were to redesign and retest—an unacceptable option that would have set the program back over a year—or accept the risk and launch as is, gambling that the loads would not be that high. Of course, before selecting the second option, we needed to assure ourselves the risk was acceptable.

We began a series of intensive reviews, which involved dozens of people and several months, trying to understand why the loads went up and whether we could believe them. The reviews involved engineers and managers from the spacecraft contractor, the spacecraft customer, and the launch-vehicle provider. The former Chief Engineer for the launch-vehicle company was called out of retirement to lead the investigation. When the dust settled, the verdict was that the loads were unreasonable. The launch-vehicle program had tried to reduce its costs and thus had only two engineers quantifying launch environments, coupling finite-element models, and computing loads for several payloads. The launch vehicle was relatively new, with only a few launches under its belt. The only way the overworked loads engineers could complete their assignments on time was to make conservative assumptions regarding data interpretation and methodology to simplify their jobs.

Look at Fig. 1-2 again, and remember the lesson it teaches: if we are to reduce the total cost without jeopardizing the mission, we must do it during the design phase. This is an essential point to understand. It leads to the following conclusion:

We can't reduce total cost—and we certainly can't ensure a successful mission—by cutting the design budget

Faster, Better, Cheaper: Can We Get All Three?

Throughout most of the 1990s, the mantra in the space industry was “faster, better, cheaper.” As a result of the recent mission failures, many have questioned whether we can get all three. I say we can.

Too few programs have truly achieved *better* because the emphasis has been on the wrong place: *faster* and *cheaper*. *Better* is not an natural outcome of *faster* and *cheaper*. When we try to do things *faster* and *cheaper*, without a sound plan for doing things *better*, things—efficiency and quality—get *worse*. But *faster* and *cheaper* are natural outcomes of *better*. If we do something *better*,

- it will be *faster*
  - simpler design, simpler interfaces
  - easier to analyze, build, and test
  - fewer problems
- it will be *cheaper*
  - same reasons as for faster
- and we’ll have a successful mission!

Now we begin to understand what ***better*** means:

- More efficient
  - people
  - processes
  - teamwork
  - communication
- Designs that are …
  - simpler and easier to predict
  - easier to build
  - easier to test
  - easier to operate
- **HIGHER QUALITY**

As Fig. 1-5 shows, reducing total cost requires an investment.

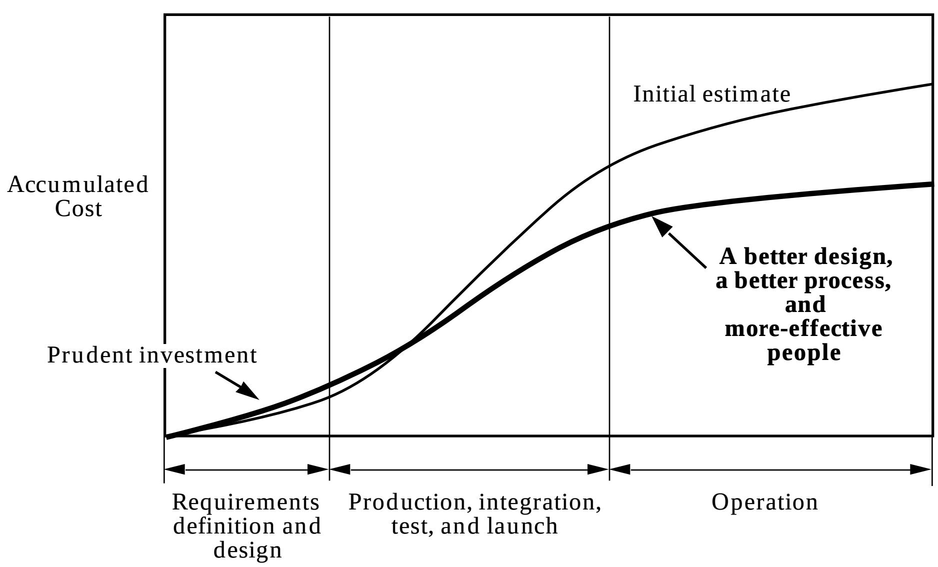

Fig. 1-5. We Reduce Total Cost—Without Jeopardizing the Mission—by Doing Things Right During Design. Responsibly reducing cost in the space industry can’t happen by mandate. It requires an investment.

There is a dilemma here, of course. If we increase the design budget, do we really expect to save money? Experience tells us otherwise. Engineers tend to fill the available space; give them more time and they'll take it. Would they give us a better design, or would they simply put more elements in their finite-element models? Would we save money, or would we lose it, with only more significant figures in the analysis to show for it? Let's be honest, here. If you were the program manager, would you give the design-support engineers more time and money?

I sure wouldn't. At least not until after I had invested in my people. Giving more time and budget to overspecialized engineers does not work because they don't understand enough about the big picture to use those resources wisely. And it's easy to become overspecialized in our industry.

Let me share an example. I have a fair amount of experience in static structural testing during my days as a stress analyst at Martin Marietta. In the mid 1980's, I was assigned the responsibility for planning my first test, which was for a large spacecraft structure. I spent a great deal of time devising load cases to bound the launch environment. But when it came to defining requirements for instrumentation, the process I used was to ask the guy in charge of the test lab, "How many strain gages can your equipment read at the same time?" His response was, "We just got a new computer system, which can handle 900 channels of instrumentation." Well, guess how many strain gages I specified? I filled the available space. The more information the better, I thought. After all, strain gages were cheap—perhaps $10 apiece. What's $9000 for a test that will cost over $1,000,000?

What I hadn't thought of was the two hours per gage it took for an instrumentation technician to measure, prepare the mounting surface, install the gage, put labels on each end of the wire, solder the wire to the gage and to a patch clip, patch the wires, and check to make sure everything worked. My flippant decision ended up costing over a man-year of labor.

Did I use all the data we collected in that test? Of course not. Five project engineers, including me, monitored data. Each of us could pay attention to about ten strain-gage readings—fifty total out of 900. I figured that was all right; we would have plenty of time to digest all the data after the test. We ran a dozen load cases, each consisting of three runs to different load levels, and the data-acquisition group faithfully recorded every channel of data at perhaps twenty load increments for each test. That's 900 x 12 x 3 x 20 = gazillion (648,000, actually) data points—and I had insisted on printouts! To interpret all that data, I would still be there today.

How much better it would have been if, before specifying locations for 900 strain gages, I had followed a disciplined engineering process:

- Define objectives (What do I need to know?)
- Consult the available experts (instrumentation technicians)
- Consider cost, schedule, and other constraints
- Find the best design (instruments and applications)

Now recognize the message here is not that you don't put a stress analyst in charge of a structural test. Experience such as this is what broadened my knowledge and understanding so that I could be more effective in the future. At many of the organizations I visit as a teacher, stress analysts are far more specialized than I was, subdivided into computer analysts, hand analysts, and fracture-mechanics specialists and having little, if any, involvement in testing. We can't hope to reduce cost if we fill our program with people specialized to this degree. As we get more and more specialized, we recognize less and less how our decisions affect others on the program. The problem expands as specialists become managers.

One popular management philosophy is that someone can manage a technical process without understanding the technical aspects. All that person needs is effective management and people skills, the theory goes. One young manager in a course I gave, during a break, said he didn't need to know the stuff I was talking about, which had to do with some technical aspect of product development. I asked him, "How can you manage a process without understanding the process?"

"I don't manage the process," he replied. "These people do." And he pointed to the engineers in the class.

I responded, "Then who is it that cuts the design budget?"

* * * * * * * * * * * *

So, because we don't know what else to do, we keep reducing the design budget, even though doing so has no hope of success. We know in our hearts there is no short cut to reducing cost; it takes better engineering, better management, and better cooperation. But how do we do that? Clearly we don't understand the problem well enough. Everyone on the program—managers, engineers, technicians—must improve their understanding of the problem and broaden their knowledge if we are to have successful missions at reduced cost.

The subject of the next article will be understanding the problem.

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