Controlling Costs - The Critical Challenge

by John Hraster
Global Aerospace Science, Goddard Space Flight Center

A recent GAO report said major NASA projects cost an average of 77 percent more than their initial estimates. More than a third cost more than double the estimate. The reasons given were technical problems, budget constraints, redesigns, and other factors. Even allowing for disagreements on the baseline–for example, was the initial estimate a Phase A study or the in-house estimate for the Phase C/D baseline?–this is still a lot of money.

Although it may seem like a cliche, the most important factors in cost control take place before you get into implementation. If the *requirements* are not realistic or well defined, you are automatically set up for problems. They must be minimized to achieve the mission; anything beyond that will add to cost. *Planning* the project involves matching the plan to the requirements and again minimizing when possible. A lean, close knit group is the way to go. A lot of people not only cost a lot but think of other ways to spend money! *Estimating costs* must be done with realism. At this stage it is very easy to forget about the cost of parts qualification, safety requirements that add cost, shipping containers, etc. There is a strong temptation to wish some costs away so as not to endanger a program that is just starting. Don't.

When you reach the implementation phase, the groundwork should have been laid, you've set the requirements, planned the project, and estimated the cost. However, there will be holes, the environment will change, and, above all, there will be problems. How you handle problems with the small amount of discretionary funding you have is the essence of cost control in this phase. Following are a number of shared experiences, thoughts, guidelines, etc., that have worked in the past. The message I want to get across is how important cost control is and how it is woven into the fabric of the project along with all the technical factors.

1. The project manager must be a leader first and manager second. This means staying focused on the goal and making sure everyone else is focused on the same goal. It is often necessary to make tough decisions now that will avoid costs three years from now. Remember, most projects fail because of poor management rather than because of technical problems.

2. It is absolutely necessary to be able to say "no.' This means in all directions; e.g., to Headquarters, to the scientists, to the spacecraft engineers and contractors. Sometimes things look like they can be done for almost nothing–don't believe it.

3. Requirements creep must not be allowed. You're buying the best product for the money, not the best one that money can buy. If you're going to advance the state-of-the-art, do it openly, up-front, with money specified for that purpose, not contingency money.

4. Treat weight, power, and computer memory as resources just like money. These items translate directly into dollars if they get out of control. A strong systems management function is necessary throughout the program to assure requirements are met and don't creep upward.

5. Know what is going on at the contractor's plant. In-plant reps help, but frequent visits by the observatory manager and others are essential. Your team members need to be intimately involved with the work on the floor and make their own assessments of staffing, shifting, schedule, quality, etc.

6. It is essential the project financial manager, observatory manager, and instrument systems manager work very closely with one another. The technical managers must understand how budgets are constructed and contingency funds allocated. The financial manager must know the technical risks, assumptions, and bases of estimate for all in-house estimates. These people should sit down regularly to match the technical and resource requirements.

7. A basic ingredient to successful cost control is a technically strong, self-confident project team. In the best case the contractor will respond with an equally strong team. The desirable result is a mutual respect of the other team's competence, which makes it much easier to reach technical solutions through compromise.

8. Contingency funds are the only discretionary funds you have. They are to be used to cover

problems within the scope of your work, i.e., to meet your basic requirements. They are not meant to cover improvements, new requirements, etc.

9. Do the work as expeditiously as possible. Future work is always more expensive. It is especially expensive if it is delayed from a present plan.

10. If it is necessary to cut back or restructure the program because of severe problems, you can't depend on increased efficiency to solve the problems. Work content must be removed. The amount of money saved by removing work will always be less than it was when the work was put in. Early planning includes preparation of descoping plans that can be implemented if you have serious trouble later.

11. Streamline the interface with the contractor. This is especially true in the reporting area. Have one set of paper do the work of two. For example, if they use a set of charts internally when their subsystem people report monthly to their program office, these can be used as their monthly report to your project office. When possible use the contractor's formats for various reports, including financial and Performance Measurement System reports.

12. It is essential to involve the project scientist, especially in the decision process. He or she is the customer on a science satellite and therefore a stakeholder in project decisions. This person must help make the tough decisions regarding requirements and cost trades, on what is critical and what is not.

13. Trust must be established between the project and the program manager at Headquarters. The program manager should be invited to all key project meetings and must be kept fully informed. This includes an early warning of problems even if they are not yet solved.

14. The award fee process on CPAF contracts must be used effectively. It is effective as a motivator only if it is honest, i.e., a 40 is a 40, a 50 is a 50, and a 90 is a 90. A consistently high score that is not deserved is not a motivator toward better performance, and it will not pass Inspector General scrutiny.

15. Do everything early. Look far ahead. Good planning can help you make informed choices when it is still possible to make them, e.g., a make or buy decision. It can also help you to decide to change a previous decision in time if circumstances have changed.

16. All the principles of TQM are appropriate to the very dynamic environment of project management. By its very nature a project is a team operation and the members must be empowered to do their jobs and challenged to generate ideas to improve the whole operation. There must be a strong interaction between the technical and business people, and between the government and contractors, i.e., no walls.

17. Dispose of problems quickly, both technical and programmatic. Although it is desirable to gather as much information as possible before making some decisions, often that is not possible. Often a non-optimum decision or even a wrong one made in a timely manner is preferrable to a delayed one. A wrong one can be reversed. If one is delayed too long, the worst case is paralysis and nothing gets done.

18. If changes are necessary to the contracted work they should be agreed to and discussed up front with the contractor. There should be no surprises in the change proposals.

19. Communicate! Communicate! Communicate!

20. Take appropriate risk. The operative word is appropriate. Appropriate risk is obviously different for manned and unmanned missions, and for A, B and C class unmanned spacecraft. Good engineering and quality practices must also be used. However, having said all that there is not enough money in any project to cover all risks. Good engineering judgment must be made in many cases when some uncertainty still exists. Look for functional redundancy as well as planned redundancy. Consider other ways of operating rather than building a "perfect" system.

Setting Requirements

A Panel Discussion

Mark Craig led the panel on "Setting Requirements," joined by Glenn E. Cunningham, Project Manager of the Mars Observer mission at the Jet Propulsion Laboratory, Larry Caddy of Marshall Space Flight Center, and David Sudduth from NASA Headquarters.

Mark Craig told the small group that a fundamental problem in setting requirements is a long development time. In fact, he said, "the longer the development time, the greater the cost," and the greater the risk of detrimental external changes, especially from the White House and Congress.

The bottom line, according to Craig, is to establish an effective system engineering process for Phase B. After all, he pointed out, "80 percent of cost is determined by the first 10 percent of decisions." Furthermore, changes in Phase C/D may cost hundreds, even thousands of times more than changes made in Phase B. "Allow no NASA requirement changes once the contract has been set," he advises.

Requirements Tools

by Glenn E. Cunningham
Mars Observer Program

Although sometimes referred to as a necessary evil, requirements are probably a project's most important element. Typically, a manager is the most concerned about cost, schedule and performance; however, the single item which affects all three aspects is the set of a project's requirements. Thus, attention to the setting and maintaining of "good" requirements should be foremost in the project manager's agenda. There are number of tools currently available that aid in the uniform generation, cataloguing, and traceability of requirements.

At JPL we started in the early 80s the in-house development of a requirements capture and hierarchy management tool called TRACER. While we clearly saw the value of the tool, its implementation and acceptance by the project community had mixed results.

As our manager of spacecraft system engineering put it, "We created a germ cell and the antibodies killed it." We designed a tool for the idealized top-down system engineering situation. It captured requirements in a uniform manner, forced quantitativeness, forced the establishment of verification requirements, and provided hierarchical traceables. All the right things. But the problem is–and I suspect that this occurs with most real projects–that we have a lot of bottom-up effects with technology constraints and inheritance constraints. In addition, we found that the people assigned to write requirements were generally more senior, more experienced people who sometimes did not have the computer skills that younger, less experienced people do, and thus we had an acceptance problem in using an automated tool with its attendant structure. We had a user friendliness problem too. Most problems were in penetrating what we call the Level 2 requirements, the mission requirements. It worked better with the hardware requirements, and exceptionally well with the design verification requirements.

But curiously enough, it has been reported that the tool has found good acceptance in the DoD community through distribution by COSMIC. We suspect that this is because there is more formality and structure in DoD's requirements hierarchy than in NASA's.

However, we still believe that automated requirement management through similar types of tools is the way of the future. The key is probably how to apply them. "Faster, better, cheaper" implies less emphasis on "how to do it" than on the top-down "what to do" and thus on requirements that are capability driven. Let the tools evolve their user friendliness, and get the younger, more computer literate people involved.