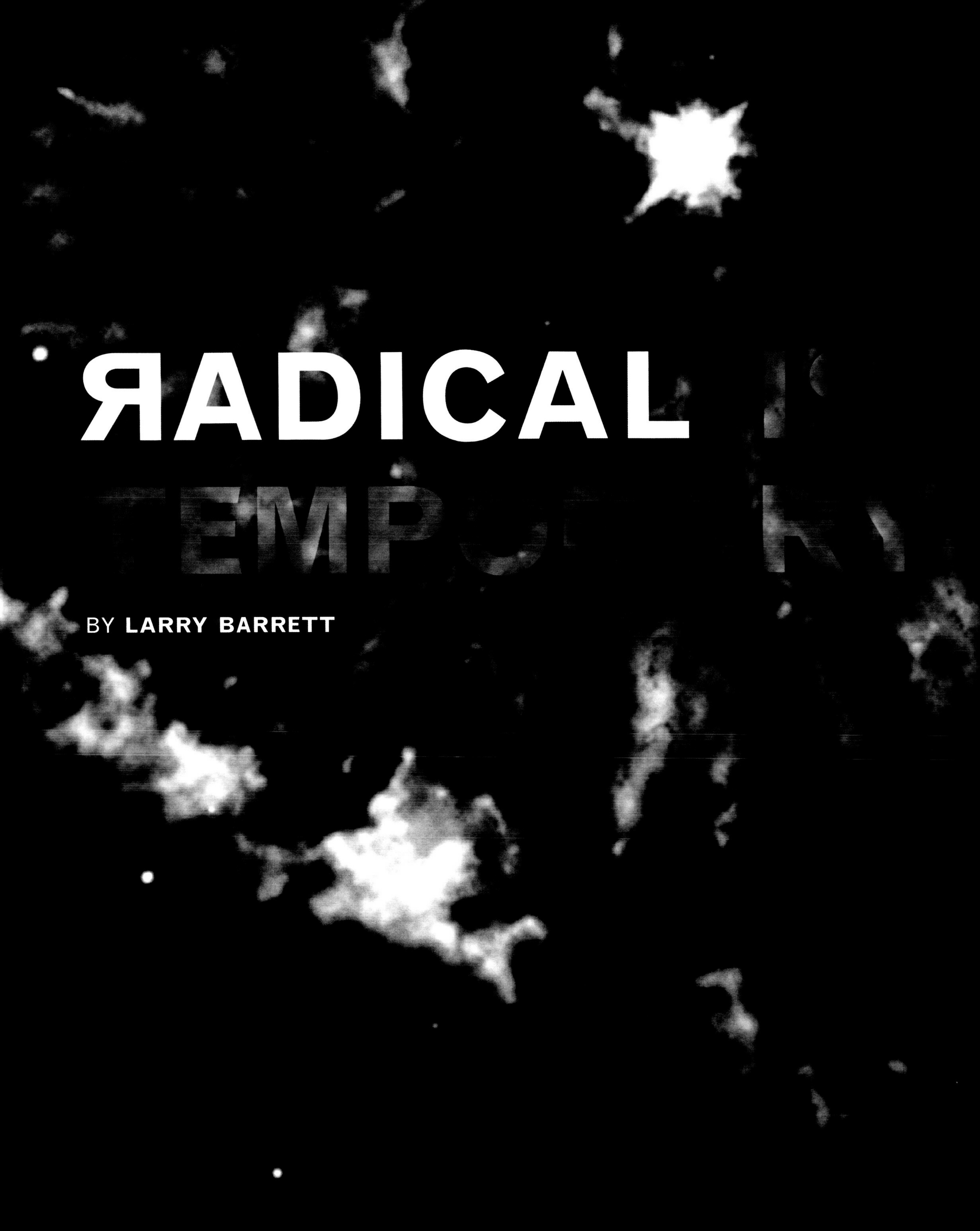


# ЯADICAL IS TEMPORARY

BY **LARRY BARRETT**

I SUSPECT THAT EVEN IF KEN HAD STAYED ON, WE WOULD HAVE EVOLVED TO THE STATE WE'RE IN RIGHT NOW. IN TERMS OF THE NATURE OF THE WORK THAT WE'RE DOING, WE'VE GONE FROM DEVELOPMENT TO MAINTENANCE, AND SO THE PROJECT TEAM NEEDED TO EVOLVE TO REFLECT THAT CHANGE.

The analogy I use to describe Hubble is a data factory, and we provide the factory controller. The telescope takes in light and produces pictures, and we're the ones sending all of the control signals and monitoring the temperature, power, and voltages in the factory to make sure the production line is doing its job and that it's not reaching some sort of a stress point. That's basically it. We maintain the Hubble command and control system.

I think we're still putting out a good quality product. We still meet our schedule and cost milestones. Every time we make a change to the ground system, we run a suite of tests to make sure that the system still runs as expected, and that it correctly controls the spacecraft. Other than that, provided the interfaces are controlled, everything is okay.

### BUT BEFORE THIS

As Ken was saying, we achieved a remarkable level of productivity and quality during the time we developed the new code. In my experience, it was exceptional—and it was something I hope to see repeated.

What made it work so well? For one thing, we had a stakeholder who decided that Hubble needed a new ground system, and she was willing to do whatever it took to get it done quickly. To achieve this goal, she was willing to allow Ken to run things the way he wanted to, including demolishing a hierarchical decision structure. From my perspective, any project demands a bounty of decisions to be made in a proximate order. What we were trying to do on this project was to get those decisions made not only well, but also quickly.

All swords have two edges. In the flat organization you can get decisions made quickly. Sometimes you are missing information and have to go back and unmake them, but in the long run I think you still save time. This is definitely the way to go when speed is paramount. In a hierarchical organization, decisions have to go through two or three levels of management to get approval. You tend to defer decisions as long as possible so you get the best answer with the most information. It takes longer, but by the time the decision is made there is usually no doubt that everyone has had a chance to comment.

Under Ken, instead of taking days or weeks to walk up the chain-of-command with a here-is-our-recommendation presentation and to walk back down with a here-is-our-answer document, everyone who had an interest in the selection of this capability, or this software product, sat down at one meeting and said, "Okay here is everything that we know. Here is how we want this thing to work. Here is how it fits in the system." In a two-hour meeting, an Integrated Product Team of ten to fifteen people could come together to make key project decisions.

Before Ken, I recall people quitting the project because of the lack of progress. There were several conscientious and technically competent people who couldn't deal with the lack of progress—feeling stalemated or blocked in our attempts to move forward. The consultant who was leading the effort had assumed absolute control, to the point that individual initiative was actively discouraged.

Another reason for our change in productivity, I believe, was that the culture of the organization was completely revitalized when Ken took over. Meetings

WE JUST HAD TO BEAR DOWN AND DO IT, AND THE ONLY WAY WE WERE GOING TO GET THERE WAS BY WORKING TOGETHER.

were non-confrontational. Ken worked to make sure they weren't. Questions came up, but there were fewer hostile challenges, like "Why the hell did you do that?" The questions were more along the line of, "Well, what else did you look at? Did you consider this?" This cultural change wasn't an easy thing to do, since it is always easier to be a critic than a contributor.

We had one guy, in particular, who was an excellent engineer, but who loved to play devil's advocate. People like that can play a useful role on a project, but he simply came across as arrogant. People didn't want to talk when he was at a meeting. He impeded decision-making unintentionally, I believe, by intimidating people into not expressing their views. If you have someone who is constantly challenging a decision, you slow the process down. As a result, Ken had him removed from the project, which was probably the right thing from a productivity standpoint. The skill was there, but unfortunately his personality was damaging to the group effort.

Ken didn't allow any one individual to stand in the way of getting the job done. We were in a phase where we knew what we had to do: reengineer an existing system. We just had to bear down and do it, and the only way we were going to get there was by working together.

## ONE PHASE ENDS

It is the nature of project work that teams evolve and move on. As new development slowed, our budget and staffing were reduced, and we went from 150 people to around 40. A lot of the top performers gradually left the project. With the technical challenges on the project diminished, the need for creativity was no longer paramount. You can't keep highly enthusiastic people around if there's not enough for them to get excited about. Many wouldn't have been happy in a maintenance mode anyway.

In the transition from development to maintenance, we also ended up losing many of those exceptional characteristics of the project that enabled our high decision rate and productivity. Had Ken stayed around, we might have retained, who knows, more functionality in the system. As it stands, we're still doing some technical upgrades because changes in the ground system are needed to support servicing missions and technology keeps changing, too. We try to fold in some new products and new capabilities, as well as implement some elements that were deferred earlier in the project because they were too costly. (Today, products exist that have made some of our former wish-list items feasible.) In a few cases, products we originally used in the system are no longer supported and must be replaced with current technologies.

As Ken said, when our major stakeholder retired, the new stakeholders didn't have the same goals as the old stakeholder. They weren't willing to accept the risk of keeping a radical project management approach in place. We all have our comfort zones, and it takes a great deal of courage to work outside of them. In all fairness, "radical" was understandably less acceptable in their career paths than it was in the career path of our former stakeholder, who knew that her next career move was retirement. We were lucky to have such a stakeholder in place at such a critical phase of the project's life cycle. Could we have accomplished what we did without the radical changes to our management structure? I don't think so.

We were on an aggressive schedule in development and, in response, we took aggressive steps to achieve our goals. A radical management approach may be something you can only sustain temporarily. But I think the results that came out of our experience on this project demonstrate the potential impact of adjusting management style to suit the real-time demands of a project. Our real challenge is making that possible. •

### Lessons

- During a project life cycle, you must examine and question what management approaches are appropriate in the current phase.
- To get maximum value out of meetings, make sure that the tenor of the group is cooperative enough so that everyone feels like they can express their views.

### Question

*For what type of decisions would you prefer a flat organization with quick informal processes?*

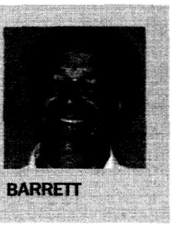
OUR RESULTS DEMONSTRATE THE POTENTIAL IMPACT OF ADJUSTING MANAGEMENT STYLE TO SUIT THE REAL-TIME DEMANDS OF A PROJECT.

## YE SHALL NOT BREAK HUBBLE

"On occasion, we would remind folks, 'By the way, this is a $2-billion national asset, and if something fails, you're going to get more visibility and more attention than you ever wanted,'" says **KEN LEHTONEN** of the Goddard Space Flight Center. Making certain that no one "broke" the Hubble Space Telescope may have been his primary responsibility—but Lehtonen was intent on accomplishing far more than that. And as these stories attest, he indeed proved to be a talented "fixer" during his tenure as project manager on the reengineering effort of the telescope's control center.

LEHTONEN

In addition to managing the reengineering of the Hubble control center, Lehtonen has served as the project lead on the development of the International Solar-Terrestrial Physics ground and science data processing systems and, most recently, as the mission readiness manager on the Aqua, ICEsat, and Aura missions. Lehtonen has more than 35 years of experience in software engineering, including 20 years of "hands-on" experience developing software applications in the fields of orbit determination, image processing, real-time data capture, and data communications.



BARRETT

**LARRY BARRETT** works for Orbital Sciences Corporation. He has more than 25 years of experience in all aspects of the system and software engineering life cycle. For the past six years, he has been the chief systems engineer for the Hubble control center system.

Lehtonen and Barrett's stories in this issue of *ASK* are not the first time the two have publicly shared their experiences working together on the Hubble Space Telescope ground system. In 1999, they delivered a paper, "Culture Management on the Hubble Space Telescope Control Center Reengineering Project," at the 30th Annual Project Management Institute Seminars and Symposium, and earlier in 2004 they published an article, "Managing a Product Development Team," in *Program Manager*. Their stories in *ASK* were based on an August 2003 presentation at the APPL Masters Forum.

Lehtonen can be reached by email at ken.lehtonen@nasa.gov, and Barrett at lbarrett@hst.nasa.gov.