RIGHT ON TIME, RADICALLY

BY KEN LEHTONEN
NASA Goddard Space Flight Center was no exception to that rule. Some folks in upper management wanted to take advantage of this new paradigm and they turned their attention to the Hubble Space Telescope ground system. The objective was to reduce the operating cost of the system by at least 50 percent. This was a noble objective, as Hubble would likely be around for another ten to fifteen years at least.

When I first was approached by my branch head with the opportunity to lead this reengineering team, I said, “Well, that sounds good. I’ve done one of these before, and it sounds like a good challenge.” So, she put me on the project. What she didn’t tell me was that I was the third person to lead the reengineering effort. The people before me hadn’t seen much success.

When I came on the project, I spent some time getting a feel for the place and what the major issues were. I didn’t try to reach conclusions about the hard decisions facing me right off the bat. Though people introduced me as the new project lead, I tried to stay in the background at first. This gave me the opportunity to observe what was going on and think about how I might need to change things.

The first thing I noticed was there seemed to be a lack of cohesion, or “culture.” We didn’t have any government on-site management; we had a consultant running the day-to-day activities. This person knew what he wanted to do, but he didn’t seem to know how to go about it. We had a number of prototyping activities underway, but they seemed unfocused. In fact, it seemed more like a technical playpen than a project. The only schedule we had was a February 1997 deadline for completing the system update before the next Hubble servicing mission. Overall, the project simply needed better structure, methods, and processes.

Though we were tasked with reengineering an old system, the project team largely consisted of people left over from the original system development. At that time, Hubble senior management felt that the reengineering effort could be completed with the legacy staff alone. This assumption later proved to be erroneous.

This is what I inherited in March of 1996 when I came on board.

Projects Are People

Yes, we had technical issues to address, but I concluded that I needed to concentrate first on the team itself if we were going to succeed. That was a strength that I think I brought to my role as project manager. In my work on past projects, I felt that was where I had contributed the most, and I knew that I had good technical people on this project who would handle that side of the house for me.

The project team had an alarming rate of attrition. I realized quickly that people were leaving out of frustration because they sensed a lack of direction. They felt that the management style of the consultant who had been in charge was obstructing, rather than enabling, work. One of the first things I did was to fire him.

I was able to convince our primary stakeholder that the project wasn’t going to succeed with just the legacy people we had in place. She said, “Fine—go off and hire some new people; do whatever it takes.” I managed to bring in about fifteen people who had worked for me in the past, which allowed me some flexibility to start to fold and mold the project the way I felt it needed to be
in order to get our work completed. The remainder of the revamped staff was provided by existing Hubble contractors, cold interviews, and “word of mouth.” At the peak of the project, we had over 150 people from fifteen different companies, plus NASA civil servants. It was a diverse group of people—from old to young and everything in between—and we were co-located, which was a good idea.

I wanted to create a “badgeless” team. I know the word gets thrown around frequently, but we took it to extremes. Since we were co-located away from the main NASA Center, Goddard, it was easier to do. We were able to have people working together who hadn’t worked together before because they were from different contractors. In a couple of cases we even had government people reporting to contractors. In the past, their management had said, “You can’t work together.” Well, we let them work together. That was a start. However, it wasn’t achieved overnight and took a lot of energy and convincing by my management team and me before it stuck.

We also flattened the organization. We got away from the hierarchical approach. We developed a series of product development teams, who were tied into the architecture that we developed. We then developed a work breakdown structure to start putting some process, structure, and schedules in place. The rewards quickly came; we started to look and feel like a cohesive project.

As we did this, I walked around and talked to the people working on the project, so that I could find out what they needed. Budget, schedule, and technical issues were all-important, but what often gets overlooked is how you get a team to work together. How do you create order out of chaos? I hoped we could create, over time, a tight-knit community much like the old Cheers slogan, “A place where everybody knows your name.” One of my earliest initiatives to accomplish this was to have biweekly barbecues, which allowed folks to have a place to unwind a bit and to talk about things that had nothing to do with work. The idea was that in six months, when they would be delivering key components in a stressful integration environment, there would be an esprit de corps to carry us through those difficult times.

Another of my initiatives I called the “kudos” program. After each major release produced by the team, I made a trip to my local grocery store to stock up on about twenty boxes of Kudos® bars. Then, I went to each individual personally and congratulated him or her on his contribution to our work. I would do that for all 150 people. This became something of an “end job,” if you will, or an in-process, as far as the relationship that I had with my team. In fact, people started bringing me coupons for the next round of Kudos that I would be buying.

I wanted to create a “badgeless” team. I know the word gets thrown around frequently, but we took it to extremes. Since we were co-located away from the main NASA Center, Goddard, it was easier to do. We were able to have people working together who hadn’t worked together before because they were from different contractors. In a couple of cases we even had government people reporting to contractors. In the past, their management had said, “You can’t work together.” Well, we let them work together. That was a start. However, it wasn’t achieved overnight and took a lot of energy and convincing by my management team and me before it stuck.

We also flattened the organization. We got away from the hierarchical approach. We developed a series of product development teams, who were tied into the architecture that we developed. We then developed a work breakdown structure to start putting some process, structure, and schedules in place. The rewards quickly came; we started to look and feel like a cohesive project.

As we did this, I walked around and talked to the people working on the project, so that I could find out what they needed. Budget, schedule, and technical issues were all-important, but what often gets overlooked is how you get a team to work together. How do you create order out of chaos? I hoped we could create, over time, a tight-knit community much like the old Cheers slogan, “A place where everybody knows your name.” One of my earliest initiatives to accomplish this was to have biweekly barbecues, which allowed folks to have a place to unwind a bit and to talk about things that had nothing to do with work. The idea was that in six months, when they would be delivering key components in a stressful integration environment, there would be an esprit de corps to carry us through those difficult times.

Another of my initiatives I called the “kudos” program. After each major release produced by the team, I made a trip to my local grocery store to stock up on about twenty boxes of Kudos® bars. Then, I went to each individual personally and congratulated him or her on his contribution to our work. I would do that for all 150 people. This became something of an “end job,” if you will, or an in-process, as far as the relationship that I had with my team. In fact, people started bringing me coupons for the next round of Kudos that I would be buying.
had the trust, confidence, and openness to stop in the 
hallways to discuss problems and make decisions 
without having to worry about any repercussions if 
they didn't pass everything through their management 
team each time.

An interesting side note to all of this was that over 
time the vocabulary of the project changed. Initially, 
there was a lot of “I” and “you.” Over time, we noticed a 
subtle shift in the vocabulary to “us” and “we.”

Not only did we meet our original milestone, but we 
had five major releases completed on time and on 
schedule. During that period, we delivered over one 
million lines of code. We were producing something 
on the order of fifteen lines of code an hour, where the 
accepted norm is closer to five. Our defect metrics were 
a third of the normal industry rate. This seemed great to 
me, but I wanted to be certain these metrics were real.

I went to a group at Goddard who study software 
development. I asked them to take a look at the code we 
had produced. They spent a couple weeks analyzing our 
work, and came back and said that our team’s work was 
some of the best they’d ever seen. So, technically, we 
were in good shape, which was what I figured.

What about programmatically? Maybe there is a 
direct correlation between high technical productivity 
and the type of organization and team that you put 
together. That would be nice to know. I got hold of 
another group who was putting together a project team 
development survey. I said, “Why don’t you take a look 
at our team? Come on over and give the survey to our 
team.” We did that for a couple of days. They came back 
and said, “We have ten major criteria for very high 
successful teams. The average that we’ve seen so far is 
about three. Well, you guys have seven.” Even program-
matically, we were off the charts.

**MANAGING EXPECTATIONS**

Despite all of the success that I’ve talked about, in 
August of 1998 I was replaced. We had a release due in 
June of that year, which we delivered on time and on 
schedule. Two months later, an organizational chart 
appeared, and I was gone.

What happened is that the original stakeholder— 
the key supporter of the “radical management” 
philosophy—retired. New stakeholders came in, 
including a new program manager who had a 
background in management rather than systems. I 
didn’t think much about the change in stakeholders 
until the day my new boss came in and said, “We’re 
going to review why you have failed to deliver; the 
project is now in stand-down mode.” Evidently, there 
was a disconnect between what we had been asked to 
deliver by our former stakeholder, and what our new 
stakeholder expected to find. The mistake or certainly 
the lesson I learned here is that one needs to continue 
to manage expectations to next-generation stake-
holders, and to do it right away. Don’t assume that they 
know what you know.

Perhaps I could have done a better job in presenting 
the case for our project team to the new stakeholders. If 
I had done a better job bringing my key people in to 
meet the stakeholders—presenting what we had done 
to-date, what the challenges ahead were, how we had 
accomplished what we had, and how effectively we 
worked—perhaps they might have had second thoughts 
and would have allowed us to continue.

I’m not convinced that would have helped, though. 
It was clear that their expectations were very different 
than my expectations. They wanted to go back to the old 
way of doing business, one they felt comfortable with, 
specifically with the prime contractor managing a more 
traditionally structured project team. If the change had 
occurred a few months earlier, it would likely have had a 
devastating effect on productivity levels—but the change 
came when our initial development phase was almost 
completed.

In retrospect, I can see that the project had reached 
a point where exceptional productivity wasn’t the 
highest priority anymore. We’ve all heard, again and 
again, that you have to know when things are good 
enough. It’s true in engineering—we don’t put twenty-
nine bolts in where we need twelve—and it’s true in 
project management.

**LESSONS**

- Nurturing a collaborative culture on a project can 
go a long way towards achieving tangible costs and 
schedule results.
- Manage expectations, not only from the people 
working for you, but for the key people, i.e. stakeholders, 
that are above you.

**QUESTION**

When would you prefer the collaborative leadership style 
depicted here, and when not?