

**NASA JOHNSON SPACE CENTER SPACE SHUTTLE PROGRAM  
TACIT KNOWLEDGE CAPTURE PROJECT  
ORAL HISTORY TRANSCRIPT**

JOHN F. MURATORE  
INTERVIEWED BY REBECCA WRIGHT  
HUNTSVILLE, ALABAMA – MAY 14, 2008

WRIGHT: Today is May 14, 2008. We are in Huntsville, Alabama to speak with John Muratore, who served in a number of leadership and management positions during his 24 years with NASA at the Johnson Space Center [Houston, Texas]. This interview is being conducted for the JSC Tacit Knowledge Capture Project for the Space Shuttle Program. Interviewer is Rebecca Wright, assisted by Jennifer Ross-Nazzal. Thanks again for stopping in to visit with us today, and as I mentioned, we know that you were with NASA for a number of years. Tell us how you first came to work in the Space Shuttle Program.

MURATORE: I was an engineering student in the senior college. In college, I took a year off and I was studying marine biology, and I was working at Woods Hole [Massachusetts] at the Marine Biological Lab. I heard about the first flight of the Shuttle being broadcast in '76. This was the first flight off the back of the [Boeing] 747, the Approach and Landing Test. I saw that happen, and as I saw the airplane, with the Shuttle Orbiter sitting on top of the 747, I said, "You know, to be an engineer and make that work, you've really got to know what you're doing." Right at that moment, I decided to switch from biology to engineering. The Space Shuttle flying off the back of the 747 was the seminal moment of me deciding to become an engineer.

So I went back to college, got my degree in engineering, and was looking for a plan out. The Air Force was just building up the Vandenberg launch site [Vandenberg Air Force Base,

California] for Shuttle at the time. I was graduated in '79, so we were still two years away from the first space flight of the Space Shuttle. I went to the Air Force recruiter and said, "I want to work in the space business." He said, "What's your background?" I said I was an engineer, and they said, "Sit right on down." They guaranteed me a slot at Vandenberg, so I went to Vandenberg for two years, and I hadn't been at Vandenberg a week and half when they sent me to the Kennedy Space Center [Florida] for the first time to get familiar with the launch system for Shuttle. I wound up going back and forth through all of the testing of the Shuttle part of the first launch. I was on console at Kennedy for the first launch of the Shuttle and all the major tests leading up to it.

After about two years, I sought an assignment at Kennedy Space Center, where I worked in launch checkout of the Shuttle as an Orbiter project engineer. I worked for some great guys. I worked for Bob [Robert B.] Sieck and Gene [James A.] Thomas, Charlie [Charles] Mars, Walt [Walter T.] Murphy, all real giants in the program. I sat console for STS-2 through STS-5. Then I went over to the Air Force side and I worked as a test conductor for one of the payloads in the Shuttle. I wanted to come to Johnson to work in flight operations, because I was very interested in being a Flight Director. I was very interested in communications and flight software and flight dynamics. In those three disciplines, not much is done at the Cape [Cape Canaveral, Florida]. That's mostly done out of Houston.

The Air Force had a contingent at Houston, but unfortunately, for medical reasons, I couldn't be moved to that. While I was in the Air Force, I had become diabetic, and they wouldn't let me be a flight controller on the Air Force side. But at NASA, Ed [Edward I.] Fendell, who was head of the comm [communications] section, offered me a job as an INCO

[Instrumentation and Communication Officer], so I reluctantly left the Air Force—I really loved the Air Force—and moved to Johnson. Ed Fendell was my first boss, working as an INCO.

It was a wild ride after that. First as an INCO, then I was called a Special Assistant for Data Systems. I was putting in prototype workstations in the [Mission] Control Center. Then I headed up Shuttle Flight Software Production Division, also called the Reconfiguration Division. Then I was a Shuttle Flight Director, and I went from there to heading up the Mission Control Center Division and leading the Mission Control Center Redesign and Implementation. Then I went to the X-38 Project and I led that for seven years.

When that was canceled, I went to University of Houston. I was working on my doctorate when Space Shuttle *Columbia* [STS-107 accident] happened. When *Columbia* happened, I came back and I headed up the Fault Tree team for the investigation, and then Bill [William W.] Parsons asked me to head up Systems Engineering and Integration [SE&I] for Shuttle for Return to Flight. He knew that was going to be a big challenge area. My title was Head of SE&I, but really I was the Chief Engineer for the Program.

After the first flight, after Return to Flight, Wayne Hale asked me to be the Chief Engineer in title. I did that for six months, and then I left the Shuttle Program. We had a little bit of a disagreement about how to proceed with the program. I went and I taught at Rice [Rice University, Houston, Texas] for a year and a half or so, and then I decided to retire. I came here, and now I'm teaching Systems Engineering, Flight Test Instrumentation, and Flight Test Data Processing at the University of Tennessee Space Institute [Tullahoma, Tennessee, UTSI]. You asked me, "How did I come to NASA?" and I've given you my whole story.

WRIGHT: That was great, because that moves us into the questions that are related to all that. I was going to mention that you worked in so many different positions, each having their own challenges. Share with us some of those challenges that you encountered on some of those positions, and the lessons that you learned that you have learned how to apply other places.

MURATORE: There were different sets of challenges in each of them. I've been really blessed in that I got to be in different parts of the program. I got to be in Launch Operations at the Cape, I got to be in Payload Operations with the Air Force, I got to be in Flight Operations with Mission Ops [Operations]. Then I got to be in the production side, in Flight Software, then I got to do Facility Development, and then I got to be in engineering in X-38, and then I got to be in Shuttle Program. It really worked out well for me, and one of my real pieces of advice to young people is change jobs regularly. I always wanted to be going to a bigger or better job. It's okay to lateral. Many of my job transfers were lateral, but they were in an area that gave me very new experiences. There's the saying: "There's a difference between 20 years of experience and one year of experience 20 times." I was very blessed. I always had very different experiences in each of the jobs, and it was really fun. I was always learning, as a result, and it really worked out well.

First really big challenge was—learning to be a Flight Controller was a big, big challenge—but after that, building the first of the prototype workstations for the Mission Control Center. It's called Real Time Data System, or RTDS, and we actually took computers and put them into the Control Center and hooked them up to the data stream, and then we wrote very advanced monitoring programs that ran them. Several of the programs that we built out of that actually became the core of the new Mission Control Center, so it was really a great prototyping

activity. The challenge there was that the people who held the institutional Control Center were very different than the group that I was in. I was in the users of the Flight Controls group, and the group that was institutionally responsible for the upgrade of the Control Center was in Mission Support Directorate at the time, a different directorate. They were very old-technology based, into big mainframe computers.

This is now 1986 or '87. We were still running a mainframe computer, a big central computer with black and white displays with very limited graphics, and if you wanted to make a change to the mainframe computer, it was months of activity. You would have to write up the change, it would be months before it would be implemented, and it was a real problem.

While I had been at the Cape, I had worked on a launch processing system. When the Shuttle Program came on at Johnson, they decided to just evolve the Apollo Mission Control Center. At the Cape, they said, "If we're going to meet the turnaround rates that we need to have for the Shuttle Program, we need to have a dramatically new system with greater automation in it." Walt Murphy headed up the development of the launch processing system. At the time, it was the most advanced data acquisition and control system in the world. By far. I had a lot of experience working on that system, both as a user and as a programmer of it, and so I started to apply that and the lessons learned to RTDS.

It was a constant battle because the mainframe community wanted to bring all the workstations and the applications there underneath the mainframe complex, and we wanted to have the power and flexibility of distributed computing. Today as people look at it, you'd go, "Why would you ever do it any other way?" But at the time, it was really radical. We didn't have any money for it. We had a few people programming part time. We found that a number of the vendors of the modern workstations really liked the idea of a picture of their computer

running in Mission Control, so we got them to loan us hardware. They can't donate hardware to the Federal Government, but we got them to loan us—there's a limit and a law, 90 days or 120 days. We would bring their hardware in and we would evaluate the hardware, and we would advance the programming. So what happened was one computer would come in, we'd learn how to program on one computer, and then it would ship out, and then somebody else would loan us a computer. It went like that for years, for two solid years. The advantage being that the code got to be very flexible because we were constantly having to redevelop it.

At the end of that time, I was offered a chance to head up the Flight Software Division, and I had built up this team, and we'd been going for two years. I wound up handing the team off to a very talented young lady, Linda [A.] Perrine, and I went into the Flight Software Production Division, where I learned a lot about how all the databases of the program are managed. Then I went into being a flight director. This was all leading towards the moment when we would redo Mission Control because in all of these jobs, I was acquiring very different experiences: as a flight controller user, as a software developer, as someone responsible for all the different databases and all the different information that needs to be managed, and then as a flight director and manager. I was on console during the first Hubble [Space Telescope] repair mission. I was one of the flight directors.

I'll remember, to this moment, John [W.] O'Neill, who was head of MOD [Mission Operations Directorate] at the time, came down and he sat on the console. The mission was over, it was the last night, and he said, "John, I'm sorry to tell you this is your last shift as a flight director." I said, "Did I do something wrong, John, that I wasn't aware of?" I was really shocked. He said, "No. We need to build a new Mission Control Center, and we need to do it in

18 months. I want you to take charge." I don't think John could have done that if I hadn't had all those particular different experiences filled.

We came in, and we had to really revamp the way things were done because they were very much in a mainframe, single central computer, heavy central control. We did a lot of dramatic things, including bringing in a lot of code that had been developed in the RTDS system. We grabbed a lot of commercial off-the-shelf hardware and software. We brought the vendors in to the program—like Digital Equipment Corporation had the computers—and we said, "Okay, what can you do to help us implement this quickly?" We took the programming staffs and we distributed them out to the user communities and said, "We're going to trust you to do the development. We're going to give you professional programmers who are going to work for you, not for us, and all we're going to insist out of you is that you give us a schedule, that you meet that schedule, and that we talk about the products that you're developing so we don't have two or three groups developing the same thing." It was a wild ride, and we went from an empty building to flying our first flight in 18 months.

Through this, we developed what we call the "pirate paradigm." This got me in a lot of trouble eventually, but the idea behind the pirate paradigm was to switch the way we looked at the problem from "this is a big facility and it's our job to protect it" to "if we don't make radical change, everything's going to go away." Therefore, what we've got to do is become very effective, and we switched the risk equation—there was greater risk to not change than there was risk just of doing the wrong thing. The people who were managing the Control Center beforehand had a good set of values. They had run a great Control Center, and they tried to avoid having bad things happen. We had to switch the paradigm and say, "Hey, look, unless we

rapidly bring this new Control Center online and we take some risks in order to do that, we are going to go out of the business because we can't afford to do business."

We put a couple things in called the "Pirate's Code" that we advertised, and there were things like the importance of personal responsibility—because if you're going to take risks, everybody in the team has to be aware of it—the importance of open communication, the importance of not being paralyzed by waiting for more data or waiting for more information. Also to recognize that you live or die based on what you do. A pirate can't afford to rely on anybody else. If a pirate doesn't succeed in his raid, he doesn't eat. If he goes out to sea and his ship leaks, he can't pull up to a repair station. If he gets in a storm and he gets in trouble, he can't call the Coast Guard. You've got to rely on yourself. We built a very strong culture of self-reliance, of trying new things out. It was a very exciting and very positive time.

We came to a paradigm which we called "build a little, test a little, fix a little." Don't try to figure the whole problem out entirely. Look at the problem and break it down into its core important pieces, then attack the first core important piece. Structure your solution such that you can go in and you can determine whether or not you've really met the challenge of that one core piece and demonstrate it, and then go on and you attack the next core piece.

For example, in the Control Center, we looked at telemetry processing for the flight controllers, and making displays for flight controllers with telemetry. That was the first problem. So we took that out of the mainframe, left command and trajectory in the mainframe, and we did a very basic, simple process at first with even the mainframe feeding the telemetry out. Then we took the telemetry processing out of the mainframe into separate computers. Now the whole telemetry process was out. Then there was a planning system where we brought that out of the computer.



We built each piece, and we delivered, in a package, something that the users could use right away that would demonstrate functionality they needed, but it broke the problem down so it didn't have to have one big solution for everything. As we learned things, when we added new pieces on, we would take the time to rebuild if we had to, to make the solution general enough to solve the problem. Without throwing stones at the team that had done this previously, they had essentially been working for ten years and over \$100 million trying to solve the whole problem simultaneously. What I found was if we could break it into little pieces and we found solutions for each of the little pieces—that in four six-month gulps, we were able to bring the whole system online. The trajectory took another five or six years to move off the mainframe, but we got all the commands, the telemetry, the network monitoring and control, the planning systems, all off the mainframe in 18 to 24 months. It really worked out well.

That "build a little, test a little, fix a little" philosophy says you're going to build something, you're going to put it in a place where you can test it, and you're going to go try it out, and then you're going to find out what's wrong, and then you're going to go fix that. Then go add some more functionality to it. We took that into X-38, and for nine years at the Johnson Space Center, there had been 12 separate attempts to start a project office to build a crew return vehicle for the [International] Space Station, which was basically an ambulance or a lifeboat for the Space Station. It had become a semiannual event; there were 12 attempts in nine years. It used to be they would say, "All right, everybody, we're going to go pick another attempt at this CRV [Crew Return Vehicle] thing," and everybody in engineering and ops [operations] would run for the hills because they knew that this was going to go nowhere.

George [W. S.] Abbey was Center Director at the time. He was very impressed with what we'd done in the Control Center, and he called me in his office one day. I thought he was

going to yell at me about something else. He called me into his office and said, "How would you build a spacecraft?" I said that I'd use many of the same principles. I'd come up with a spacecraft design that we could test in the atmosphere, test in pieces, and then slowly move up into space flight. I'd start building very primitive vehicles and then build up to much more sophisticated vehicles. And I'd try to use as much commercial, off the shelf, and reused components as I could. He said, "When will you have it ready?" I said, "Well, I'm going to need money," and I went off and put a budget together. And he said, "All right, we'll get that for you." He got me about a tenth of it the first year, and that started the X-38 project.

In the X-38 project, we broke the problem of bringing people back from space into several challenges. First, to make it work, we decided on using a lifting body with a large parafoil, and the large parafoil had been a technology problem. As best we could determine, there had been ten different attempts to make the big parafoil work. We thought the Army had made it work pretty reliably when we took parafoil on. After we started testing it, we found that the parafoil didn't work very reliably, and we had a battle that took us over 20 test drops of full scale and over 400 subscale drops before we finally got the parafoil working reliably.

We did that [in] Yuma, Arizona, shoving palettes out of the back of helicopters and out of the back of airplanes. We built a vehicle in fiberglass, which we dropped out of an airplane—parachute failed and we went into the ground and made a big hole in the ground in Yuma, Arizona. We learned a lot from that. We built a second one that we dropped out of the wing of a bomber that was very successful, but it had some problems. It didn't have an active control system. It just dropped and glided and then put the parachute out. Then we built another one that had an active control system in it that could fly to different positions. Then we modified the

first one to be a more sophisticated control system. Anyway, we broke it up into a series of challenges that way. So we built a little, tested a little, fixed a little.

At every step of the way, just as in the Control Center, we had people involved in every aspect of it. If you were the battery guy on X-38, you were the battery guy who figured out what kind of batteries we needed, you bought the batteries, you designed the assembly they fit in, you put them into the spacecraft personally, with a technician. When they had to be pulled out for maintenance, you skinned your knuckles pulling them out. You sat on console and you watched them operate. You might even have sat all night charging them up. It got people very knowledgeable about the entire life cycle.

We had used that same approach in the Control Center. We didn't just have people who designed the system, or coders or testers. People took entire applications through the entire life cycle. It was very satisfying for the people because they got to see the whole thing. It avoided us having to have different little fiefdoms of reliability, maintainability, all those sorts of things.

We did have certain specialty areas for advice. We had a really good safety officer in Larry [Lawrence G.] Schmidt—had tons of experience and really helped us out. We teamed with the [NASA] Dryden Flight Research Center [Edwards, California]. They had lots of people who gave us expertise on how to assemble the vehicles. The first vehicle we sent to Dryden—they complained it was the worst vehicle they had ever seen. We spent a lot of time asking them what it would take to make it better, and they told us, and we built that into the second vehicle.

The third vehicle got to Dryden, and they said it was the best research vehicle they'd ever worked with. So I came to the conclusion that this "build a little, test a little, fix a little," along with having people involved in the entire life cycle, enabled you to learn information and incorporate it into your design. If you look back at the history of NASA, in the '60s that's what

NASA did. It built Mercury, it built Gemini—and there were several different flavors of Gemini and the Agena Target Vehicles—and then it built Apollo, all in a ten-year period. The people who were working at the end of Apollo, who designed Skylab and Shuttle, who designed the Saturn V and the Shuttle vehicle, had all been through many different design cycles.

One of the traps we're in, in the industry right now, is that we have people who have been working ten or twenty years and have only ever seen a program in one phase. They may know a lot about operations and they may have very deep knowledge about operations. Or they may have very deep knowledge about launch. Or they may have very deep knowledge about requirements or in conceptual design. But very few people have crossed all those boundaries. So by doing the "build a little, test a little, fix a little" philosophy, you get the advantage of getting to have lots of experience very quickly, and you generate things that are useful that you can use, and at the same time you go ahead and you get a lot of experience.

It's very hard to do big programs this way. However, it is possible to do that if you architect from the very start. For example, Space Station: we took the decision to go for the full-up Space Station from the very start. If we had taken more of a Russian model and built a little Space Station, and then gotten rid of it and built a little bigger one, and then maybe added onto the bigger one, and grown it that way, we might have been able to get a Station going faster and more flexibly, and we would have gotten a lot more people involved in design, and we would have been able to build it a lot faster, I think.

Similarly, you begin to see a little bit of that in Constellation right now. They're embracing some of this philosophy. They're testing the launch escape system separately from the main. They're testing the first stage of the rocket separate from the whole rocket. So they're

beginning to architect the program a little bit to take advantage of that, but you have to make those decisions up front if you're going to "build a little, test a little, fix a little."

WRIGHT: Well, if you would, that's one of the areas of discussion: the importance of planning. But on those projects that you've worked on, you had a short turnaround time, so will you share with us how important it is to plan as you go along and how you do that?

MURATORE: Yes. A couple of quotes: "A good plan today is better than a great plan tomorrow." That's a pretty management saying, but it's really true. There's a tendency to get so focused on making the perfect plan that we don't do anything. That was a real thing that happened in the new Control Center. It happened a lot in the attempts to make Shuttle upgrades. It happened to the CRV program. What I found is that, particularly in NASA, things are so dynamic because of the environment we're in, it really doesn't make sense to do a detailed plan because the environment's going to change around you as you go to it. So what's needed is to have the *right* things planned, the *right* degree of penetration of the technical issues, and then combine that with a flexible approach. Part of that flexible approach means: you need to be able to prepare it to change rapidly. That's very hard in big programs because in big programs, changes have wide-ranging impact. But you can organize yourself up to do that with a strong systems engineering function.

In X-38 we had a 200-person program, and I had a small cadre of about 12 people doing systems engineering, but basically everybody had a systems engineering hat. Everybody knew what their piece did and how it interacted with all the other pieces on the spacecraft. That was possible because we had a relatively small spacecraft: 24 feet long without the orbiter module, 35

feet width. We had a relatively small vehicle. It's bigger than the CEV [Crew Exploration Vehicle], though. It was in comparable complexity with the CEV. If you looked at CEV, they've got way more than 200 people working on it right now, and it's a very different approach. You have to be able to deal with change rapidly, and you have to have a strong group of systems thinkers constantly engaged.

The program management needs to be constantly engaged from a systems thinking viewpoint. If you look at Apollo and Gemini, you'll find that the leaders of those programs were very much that way. I was recently reading a book by [Stephen B.] Johnson called *The Secret of Apollo: [Systems Management in American and European Space Programs]*. I use it in my course at UTSI. It's a great book, and it points out that [Dr. Robert R.] Gilruth as Center Director knew every nut, bolt, and washer in the Mercury and Gemini spacecrafts, and that [Dr. Wernher] von Braun knew the details of how everything worked in the Saturn V.

Nowadays, you hear things like, "Well, let's not engineer it here." You hear that in meetings a lot where managers feel that they shouldn't get involved in the details of what's going on. I think the key to being able to deal with rapid change is to be able to deal and engage on those details, and have mechanisms where everybody in the program understands the responsibility associated with change, and where you communicate change on a regular basis. I'm not just talking about organizational change, I'm talking about technical change. That needs to be communicated very rapidly.

WRIGHT: I'm going to use your pirate analogy. How'd you keep all your pirates on the ship without abandoning through this time period?

MURATORE: It's interesting. I've talked to the people who built the Mars Pathfinder, and not everybody can operate in that mode. In Mars Pathfinder, they brought their best and brightest. Some people stayed and liked it, working that way. Some people couldn't deal with the lack of structure and left. It didn't mean they weren't good people, it just meant that they couldn't deal with a lack of structure. X-38 and the Control Center—we found the same thing. The vast majority of people could deal with it, but some people couldn't. Those people needed to find jobs within NASA that are more structured. There are important, more structured jobs we had. But the turmoil of the design phase, the rapid design, rapid change—they couldn't deal with, so they needed to go work on things [with] considerably less turmoil in them.

Given that NASA's operating in a superheated political environment—which it always has during my entire time with NASA, and I suspect it will for the rest of NASA's life—you've got to be prepared for the fact that a program that takes four, eight, or ten years to execute is going to go through lots of radical change, and you've got to be flexible and able to deal with it. The goals that you set today, you're going to have a hard time defending ten years from now. So what you've got to do is be able to deal with and incorporate change as you work your way through it.

I had people who stayed with me, who were with me in RTDS, through Mission Control Center upgrade, X-38, and into Shuttle. The people who come to work at NASA want to get their hands dirty, and they want to accomplish something. They want to get something done. So consequently, if you can establish a track record of, "Hey, I don't know what it's going to take, but we're going to really go do something here today," and all of your decision making, all of the efforts you do are all about getting the mission done, you have to beat people away with sticks. You don't have to worry about holding on to them.

One of the most precious things all of us have in life is time, and one of the worst things we have is people who waste our time. As a manager, I always felt it was sort of a sacred trust—what I would call a “blood oath.” When I took on a project, it was either carrying my shield or on my shield, we're coming out of this program like the Trojans. We're going to either succeed at this and be able to hold our heads high, or we're going to be carried off the field because we did everything, we just died trying.

People know whether you're committed or not. If your decisions reflect that kind of total commitment every day, if they see you every day trying to find solutions, trying to work through things, not tolerating any mischief and focusing on, “What is it going to take to succeed in our goals?”—and quite frankly knocking over opposition when it's appropriate to knock over—knock over it, maneuver around it, dig under it, climb over it, whatever it takes—they realize that you're not about wasting their time. I think anybody who's going to be a manager at NASA, that's the most sacred trust we get: we get really talented people, and we get their lives. Our people throw their lives into a project. There's no nine to five in NASA. Everybody who works at NASA—you find you've got your normal two percent of people who don't work very hard—but other than that, everybody works hard.

If they're not working hard, it's because they don't feel a commitment to the project, and if they don't feel a commitment, it's because they feel that the management's not committed to really succeeding. It's a really straightforward proposition. I can't tell you how many times I've seen projects where the people leading the projects weren't that committed. They were committed to maybe their own career, to the success of the Agency, to the success of politics, to a variety of things—but the thing they were doing, they were not totally convinced it was



necessary, it was important, and that it deserved every ounce of energy that they had. People sense that.

WRIGHT: What you shared with us are some really valuable lessons and insights, specifically, to those two areas that you talked about. How did these work? Share those lessons and how they worked with inside the Shuttle Program when you had your other positions.

MURATORE: Shuttle was really a shock for me. Mission Control Center was a quarter of a billion dollar program. Over 18 months, \$250 million, almost 700 people involved. So I thought I'd run some pretty big things. The X-38—maybe 300 people involved, maybe \$30 million a year. Not very much money, but we had ESA [European Space Agency] involved, we had Dryden involved, we had [NASA] Marshall [Space Flight Center, Huntsville, Alabama] involved, we had Johnson involved. A wide span of station programs. I came into Shuttle saying, "All right, what I'm going to do is modify these techniques to work in a big program." I was really surprised that they required quite a bit more modification than I thought.

What I found was the biggest problem in big programs is communication. In X-38, I used to hold all-hands meetings, and I would bring everybody into the building, I'd stand on top of a table, and without a microphone or anything, I'd just talk to everybody. I'd tell them what we were doing, and where we were going, and what the issues were. On a daily basis, I'd touch base with maybe 10 percent of the people, and rotate it every day who I was talking to. Same thing in the Control Center, 700 people. Shuttle Program—maybe 10,000 people spread over a whole country; communication becomes very difficult.

In fact, the informal communications mechanisms of the program, which is the rumor mill, the grapevine, works far better than the formal communication mechanisms of the program. There's a saying: "No good deed goes unpunished." Consequently, what shocked me was very straightforward things that I did, which I attempted to communicate in formal channels like control boards and directives, were interpreted in wildly different ways in the different corners of the program. These wild interpretations were communicated at the speed of light, whereas the formal communications were snail mail.

It forced me to think a lot more carefully about what I did. Because in X-38, I could walk in and I could say, "Oh, this design's a piece of garbage. We're going to get together and we're going to go fix it." If people got hurt, within an hour we'd be in the room trying to solve the problem; they saw I was committed to solving the problem, and the fact that I called their design a piece of garbage was forgot. Matter of fact, a couple hours later, they would be going, "Here, check this piece of garbage out." It would be very easy. Shuttle Program—so big, so diverse, so much money, so many big contractors, so many big egos, so much big politics that what happens is that if you just kind of lay it all out on the line like that, it comes back and swats you really hard.

I had to modify my techniques significantly in my communication style, in my expectations, in the jobs I sought for people, in the way I tried to institute change and how much of a change I could go after at a given time. I had to plan a little bit more. I had to spend a lot more time communicating. In the X-38, I'd say something once, and everybody would get it. In Shuttle, I would have to give the same story five and six times in order for it to sink in. I had to be aware that things would be interpreted sometimes even 180 out. I would say one thing, having one set of values, and it would be interpreted as 180 degrees out.

One of the biggest challenges in Shuttle Return to Flight was, as we know, when the bipod foam came off on [STS-]112 and hit the solid rocket booster and put a dent in it. The next Flight Readiness Review there was quite a discussion about, "Were we safe to fly?" Bryan [D.] O'Connor, who was in safety at the time, asked to see the hazard analysis. They brought it out, and it was a terribly confused document. It hadn't been updated in years, and it didn't give him any help at all. The integrated hazard analysis should have said, "Foam damage this small is acceptable. Foam pieces this big is not what we expect to happen and would represent a big risk." So one of the things that the CAIB [*Columbia* Accident Investigation Board] pointed out was we need to redo the integrated hazards.

When I came in, I reviewed the integrated hazards. I am not a big process guy, but I had learned through X-38 that integrated hazard analysis really can help you a lot. Integrated hazard analysis is where you sit down in a very systematic way and go through what could go wrong, what would happen. You start at, "What's the worst thing that could happen?" and you say, "What could cause that to happen?" Then, "What could cause *that* to happen? What could cause *that* to happen?" When you finally get down to the bare bottom of the root cause events, then you say, "What have we got in place to prevent that from happening, and how do we verify that that control is in place?"

I was a real believer in integrated hazard analysis, and so I took off to go rewrite those. I tried several mechanisms to get them rewrote. It took me six months before I finally realized that I was not working it the right way. The contractor was responsible for updating the integrated hazard analysis, and I first took them to task in their evaluation for not keeping them up to date, and then I set a series of meetings and activities to try to get them done, and we made no progress whatsoever. So I wrote an incentive into the contract to get them done by a certain

date, and I put a lot of money on this incentive. It was highway robbery. But it got the work done.

Another thing I learned is in this big business, you've got to understand the thing that motivates the contractor. And it wasn't shame that *Columbia* had happened, it wasn't pride in doing their work right, it was were they going to make a profit at it or not? When they found that they could make a profit out of doing the right thing, they worked really hard at doing the right thing.

The other thing that I learned in this process was that the Shuttle Program, these big programs, are driven by fear. The fear of being canceled, the fear of being told that you didn't deliver on your responsibilities drives these big programs inordinately and it causes them to take all sorts of risks that I wouldn't have taken on X-38. Matter of fact, what astounded me coming into Shuttle was things that if I had said on my little, unmanned X-38 test vehicle, "I want to do this," Dryden and Edwards Air Force Base [California] would have never let me do, we routinely did in Shuttle. We have a lot of bad practices in Shuttle Program from an engineering perspective. As a result, when we tried to change those practices we ran into this fear. "If you guys do this, it's going to take us forever, and we're never going to get back in the air. We're going to get canceled on the ground."

In doing the integrated hazard analysis, first I put the contractor incentive in place; that was the first thing to do. The second thing—I took a play out of the X-38 playbook—people laid out for me a schedule, and it showed that even with the incentive and everybody working hard, it was going to take us another year to get the integrated hazard done. I said, "Okay, what that means is every Saturday, we're going to come in and work on integrated hazards," and people

looked at me in shock. I said, "That's what we're going to have to do to get back in the air doing business the right way."

For 13 Saturdays in a row, all day, eight hours, we worked on integrated hazards. I took the lead and I said, "I want you to have a package ready to go every morning when I walk in on Saturday morning. I'm going to come in with a set of donuts, and I expect by the end of the day we're going to have worked through the package, that we'll have eaten all the donuts, and that we will have something ready to move on with." We did that for three and a half months every Saturday, and it cost everybody quite dearly. But we got it done, and we got an incredible product done. We had the safety people involved, we had engineering, we had operations. We had everybody involved, and we got a superior product which identified a number of other problems in the program which we went and fixed which could have just as easily led to an accident like *Columbia*. Also, the integrated hazards are now used and referred to on a regular basis in the program.

A number of the tools and techniques I used, I used, but I had to modify them. Selling the program on letting me do that, letting us do that, was quite a struggle. We had a Saturday Program Requirements Control Board where we had a violent set of arguments about whether we should take on doing the integrated hazards. Fortunately, Nancy [J.] Currie, who was Head of Safety for the program at the time, stood shoulder to shoulder with me, and we just said, "Hey, we're not going to sign off on Return to Flight unless this is done."

Bill Parsons backed us up. It was really interesting, because Bill was like, "I don't believe you're going to get this done, but I understand that you're committed to it, and I understand why it's important, and I'm going to let you guys go. But you better get this done." We got it done. What was interesting was when we were doing the Saturdays, a number of

people criticized us for working people too hard and felt that that was not safe. They were concerned about exhausting people out. Of course, this was done six months before our first flight—a year to six months. I had to argue—we're going to have meetings on Saturdays, we're going to work this problem. But at the end of it, everybody was a lot smarter.

One of the biggest advantages you have as a NASA manager, one of the biggest benefits you have as a NASA manager—I consider it by far better than anything that they paid me in the entire time I was at NASA—as a manager, if you choose to, every day can be a day you walk into a room and you learn from incredibly brilliant people. If you choose for that to be your job. I've always felt that my job was to go in and ask questions and get smart on other people. As it turns out, our people love to teach. If you go and admit, "Yeah, I'm in charge, but I don't know what's going on. Tell me how this works. Explain to me how this works," and you ask good questions, and you're an active listener, and you get engaged with them, they'll do pretty much anything for you. Then when later on you might have to do something that they don't understand, they're going to come at you from a viewpoint of, "You understand this because we spent two hours going through this, so let me tell you why you're doing something stupid right now." Then you can explain why it is that you want to go do what you want to go do.

I think one of the biggest lessons I've learned is that the best thing to do as a manager is to walk into a room and ask questions. One of the worst things that happened to me at the Cape was this feeling of "you can't interrupt." Everybody has to have their say, and everybody has to be polite, and it needs to be a set piece kind of discussion. I think that that was a real loss because the interrogating back and forth is how new information comes on the table and how you learn.

Now, you can abuse it. First of all, questions should never be abusive. They should never call into question someone's character or integrity or their knowledge, but they should be about "What's the subject at hand? How does this work? How do we know this works? What's the physics behind that? What's the chemistry behind that? What's the common standard practice that's done in the industry? How do they do it on other programs?"

Those are all great questions to ask, and we really need to grow a generation of managers who are active questioners rather than they're managing so that everybody's voice is heard—it's a very important value, for everybody to feel that they have a chance to communicate; however, that can happen inside in an environment of constant questioning going on and constant learning. I think if you can communicate, you can ask questions to communicate intimidation, or you can ask questions to communicate "I want to learn." If you're communicating "I want to learn," you can get an awful long way. We did a lot of it.

Another example of that was the interface control documents. One of the critiques were that they were out of date, and when we did a survey of them after *Columbia*, we found huge out-of-date ICDs. So I had ICD meetings where I said, "Come in and bring the change, and explain to me why the ICD is wrong, why it needs to be a different way," and then I'd approve the changes on the fly.

One of the things I did learn was that even big programs need rapid change mechanisms. It used to be the Control Boards for Integration had a very—a change would come in, everyone would have to review it and sign off on it before it would go to the main board, and I said, "No. Everybody show up at the board. You'll hear the presentation." To make a big program work, the presentations had to be distributed ahead of time. But they didn't all have to be reviewed and signed off and gathered comments from everybody. We could go talk about it around a table.

We changed some of the requirements for putting instrumentation on the Shuttle so that we could put it on more rapidly when it was a low-risk instrumentation upgrade. That was because it was simply taking too long to put instrumentation on. It was way too expensive. So we backed off some of the requirements. Basically, we did a risk assessment and said, "This is a low risk thing to put this sensor on, an accelerometer, a temperature sensor, so let's go do it," without doing all of the engineering that we would normally associate with the change. What I had to trade was the engineering knowledge we weren't getting versus the risk of not engineering the way we did all the operational equipment. What I concluded was we didn't know enough about how the Shuttle was flying.

That was one of the things that really shocked me—we knew more about how the X-38 flew than we knew about the way the Shuttle flew. One of the things I did during Return to Flight was I got all the wind tunnel models out of the storage. We updated them to the current Shuttle configuration and took them into the wind tunnels in Tullahoma up in New York, and in California, and we generated a whole new database. One of the problems big programs have is they want to make changes, and they may want to make a good change, but they get into not engineering the changes. They get into saying, "Well, if it's a small change, it'll be okay." We had accumulated so many small changes that we had no idea what the baseline was anymore, and so we had to go and break out the models and do the whole program again.

To make it work in a big program—the pirate paradigm, you can make it work. You've got to be aware that whatever you do kind of ripples through the system very quickly, particularly unintended things you might do. You can build rapid change mechanisms in certain areas. You can demonstrate your commitment to a solution. You can ask lots of questions and be known as somebody who wants to be a learner and a listener. Those things you can translate



very well into a big program. With some of the other things, like sitting in an eight hour meeting—I hate long meetings, but sometimes there's just no way to go do that in a program as big as the Shuttle Program without going through that. Like I said, it used to be in X-38, people would walk in with the charts and we'd deal with it.

My understanding is that in early NASA, it was very much the same way. We didn't have view graphs and projectors, but people would bring schematics into von Braun's conference room, pin them up on the wall, and then as they would brief von Braun, he would walk up, and they would point to the schematics. Same thing with Gilruth. We don't do that anymore. Everything is a process, and there's a review, and there's people who've got to go, and the signature and the sign off is more important in the process than, “Did we really engage on this problem? Did we really talk about it? Did we really think about it?” The processes, some of the techniques, do translate.

The "build a little, test a little" translated because we broke the Shuttle debris problem down into several problems. We had this giant debris problem, and we broke it down into, “What's the problem? How does debris come off and why? And what is it like? How does it transport through the flow field around the vehicle? Then what are its impact conditions? Then what is the impact tolerance of the materials?” We broke them into three basic mini-projects, and then we attacked each of those projects as mini-projects. Then each of those mini-projects had a part for the Orbiter, a part for the tank, a part for the SRB [Solid Rocket Booster], a part for the RSRM [Redesigned Solid Rocket Motor], a part for the ground system. Each group had their own little mini-project to go do, and so we'd “build a little, test a little” that way. As we found out more things about what the impact tolerance was, about the size of the debris being liberated,

and what the flow field did to it, then we would change our plan, and we did it in a series of cycles. So we even used “build a little, test a little” to get our way out of the debris problem.

WRIGHT: Let's talk about risk. Everything that you did, whether it was the Mission Control configuration to the projects to communicating, it all has some risk involved in it. Tell us how you thought about how to mitigate the risk and/or handle the risk after it happened.

MURATORE: The first thing I would have to say is you've got to be tough enough to not worry about labels. In my career, I have been labeled a wild risk taker, and I've been labeled the most conservative person in the program.

WRIGHT: That's quite different.

MURATORE: Yes. Same guy, same techniques. When we were building the new Control Center, the Center Director called me up to her office—it was Carolyn [L.] Huntoon at the time—and said that they got a number of reports that I was taking wild risks with the Control Center that were not safe. With X-38, I had a number of battles with the people at Dryden. They said we were taking wild risks, unnecessary risks. In Shuttle, the critique was always the opposite. It was that I was not willing to take risks, that I was too conservative. As I look through it, I think there was pretty much a constant level of engineering rigor that I put into the work, no matter where it was. As I got a little older maybe I got a little bit more conservative—the Shuttle Program was in such a pressure cooker and such a magnifying glass—maybe a little bit more conservative, but not a lot.

Risk, first of all, is about perception. You've got to be tough enough to base it on your own analysis, and not on labels. You've got to be prepared that one day, you're going to be told that you're a wild-eyed cowboy, and the next day you're an over-conservative, and you've got to just let that stuff bounce off your shoulders. Shuttle Program Managers have struggled with that in the past. It's very hard to do that. You've got to have a lot of confidence in your own estimate.

The second thing about risk is to understand the difference between reducible and irreducible risk. At UTSI, we fly. We build experiments and fly them. I've been here at UTSI five months, and we've already flown a bunch of new hardware on our airplanes. What I tell my students is to be bold about irreducible risks and conservative about reducible ones. At some point, you come to flying, and there are unknowns that you can't do anything about, at which point, be bold.

However, when you're still working out your understand of how things work, and you can do ground testing, you can do ground analysis, you can do other things like that in order to get a better understanding of it, you need to keep reducing the risks down. I think that's the largest confusion we have in risk management, what we often do is—we have a bad thing happen, we get very conservative, and we do all sorts of bizarre, extreme engineering. Then we get tired of it and go, "Okay, well, the risk to fly is the risk to fly, so let's go fly," and we've spent a lot of time trying to work on irreducible risks, and we haven't spent time working on reducible ones. I feel that that's the biggest thing we do wrong—not putting our energy into reducing the risks that are real and reducible, and we spend all of our time anguishing over the risks that are irreducible.

WRIGHT: Will you give us some examples?

MURATORE: As part of Shuttle Integration, I ran the winds aloft issues for the Program. We spend huge amounts of time worrying about winds aloft. We have very complex ways of evaluating the winds. In the end, there's no way to instantaneously measure the winds and deal with the winds. The winds are in God's hands; they're not in our hands. So all we can do is make a best estimate, project what the wind are, evaluate our capability against those winds, and then go fly. We have very elaborate processes to do that, very complex processes to do that, over what is essentially a simple problem.

A case of reducible risk that we have trouble following is wiring bundles. After STS-93, we launched, we had a short. Took down an AC [alternating current] bus, we found all sorts of substandard wiring in all the vehicles. We went through this process to go over, inspect, and repair. There were some people in the program who, with good motivation, said, "We've got to stop tearing the vehicles up. We're doing more damage to the vehicles than we're finding." Or, "The risk of damaging while we do it is greater." Turns out we were finding way more damage than we were making. We were making damage when we would tear the vehicles up, but we were finding way more damage than we were tearing up. That, to me, was a reducible risk. I fought really hard to get all the wiring inspected, and there were a lot of people who didn't want to do it. But I said—and I wasn't the only one, there were a lot of people: Nancy Currie, Bill Parsons—we needed to inspect the wiring because if we have a short due to something that we built in there and we could have inspected and fixed, that was reducible risk.

We have a problem in the wiring Shuttle called capton wiring, and the problem with that, the wire insulation—the thing around the conductor is made out of a material called capton—

and, unfortunately, if you get a short in the Shuttle wiring, you can get a short that's big enough that it can cause the insulation to ignite, but not big enough to trip the circuit breakers. Short of tearing all the wiring out in all the Shuttles, there's no way to fix the capton. I reviewed that as irreducible risk. I stopped worrying about capton.

Now, on X-38, we had Teflon wiring the way we did in Apollo. They went to capton because it was lower weight. It was the best wire available at the time. It was lower weight, and people didn't know about this arc-tracking phenomenon that can cause ignition. It's not possible to tear all the wiring out. It's just not. So my view was, "Let's deal with the reducible risk. If we have nicks that can short and cause a short, and then after we do that, stop worrying about it being capton." We can't do anything about arc tracking. That's an irreducible risk.

We had a lot of fights about whether debris was an irreducible risk or not. My view, and the view that ultimately led to me leaving the program, was that the foam risk was significantly more reducible. Other people felt that we'd already reduced it so significantly that further effort, extreme efforts in delaying the program was unwarranted. Particularly given the hard situation—we have an issue with the workforce so stressed because of the hurricane, and lots of them still living in trailers three years later, four years later. But I felt the risk was still reducible, and I felt that we could keep doing engineering to reduce it.

One of the things about irreducible risks is, in my experience, people tend to overestimate the irreducible ones and underestimate the reducible ones. In other words, they overestimate how big the irreducible risk is—"It's way more risky to do this due to factors we can't control,"—and they underestimate the things we can work on. They say, "Well, we don't need to work on that anymore because it isn't so big." Well, in fact, in my experience, it's always the things that you can reduce that bite you.

We have not lost crews because of acts of the natural environment that we can't control. They tend to be always things that we could have controlled. Sometimes you run into things you can't control. DC-10 crashes because it runs into a microburst over Denver [Colorado]. It was undetectable with the technology at the time, it just happens. But that hasn't been our experience in NASA. Our experience has been reducible risks tend to bite us in the—to quote a book in the NASA training program, a book by [James R.] Chiles called *Inviting Disaster: Lessons From the Edge of Technology*—it's used in the Seven Axioms of Good Engineering Course that I teach—they say, "If you ask the question: When has there ever been an aviation accident for which there was not a precursor warning incident?" The head of the NTSB [National Transportation Safety Board] said, "My answer would be to you, never."

Most of the risks we've got are reducible if we're willing to do the engineering and put the work into them, to go knock them down. Sometimes you can't knock them all down at one time. Sometimes you need to knock part of them down, go fly, get a little bit of experience, and then knock them down a little bit more. I can buy into a strategy that says, "We're going to knock the risks down a certain way, but we won't get them all in one. Then we'll knock them down a little more when we get a little more experience." I think there's something to be said about that strategy, which is sort of where we are with the foam.

The other part about risk management is ultimately risk management always comes down to dollars. We have very elaborate risk management databases in the Station Program, we've got one in the Shuttle Program, with hundreds of risks. But we don't have the resources to cover hundreds of risks, and I think that whole risk identification process is way overblown. I think you should have a pretty good screening process that's constantly bubbling up new risks, but

after that, you really can't mitigate them. So what I always used was what I called "Top X," which was the top ten or twenty risks that I could afford to do something about.

Risks would fall off the page when we found out about a new risk. But we're so resource limited at NASA, and in most cases, you can't even move the resources around because they go to certain Centers, they go to certain elements of the program where you don't have flexibility to move them, so you only really have flexibility in a very small amount. You only really can mitigate two or three risks, or maybe ten risks in a program, in a big program. I would spend more time trying to find the risks, and talk about the risk, and focus your resources on the top few, keeping everybody's eyes on the top few. That's what I use and I put in place in Shuttle Integration, what we called "Top X." Did that in the Control Center, did that in X-38. Once a week we'd get around, "Let's talk about what are the top ten or twenty issues that are going to cause us a problem. And are we putting the resources on those top ten or twenty issues?" We may talk about the same issue every week for the next two years. That's okay.

WRIGHT: Are there other recommendations or suggestions you'd make to improve the techniques for risk management and risk mitigation?

MURATORE: No. There's something I'm thinking about, but I'll tell you about it later.

WRIGHT: All right. Working with your X-38 team of 300 folks—everyone worked close together, you had a structure within an ever-changing environment. I'm going to make an assumption that people trusted each other within that group. How do you instill that trust and

how do you build that trust within that group so that it can trickle down and trickle over to others to get them involved?

MURATORE: I wouldn't say we started out uniformly trusting each other, and I wouldn't say that at all times everybody trusted each other all the time. As a matter of fact, there was sort of a constant "Peyton Place" of evaluating each other. You've got to understand: X-38, we literally put each other in each other's hands. I flew a bunch of tests that if my guys had done something wrong, we would have gotten in the air and we would have been dead so quick it would have been in an eyelash. Other people put themselves in my hands and other people's hands. What we did was we communicated the fact that we relied on each other. We constantly communicated. We constantly made each other aware of it. That's very hard to do in a big program.

Glynn [S.] Lunney, the famous Shuttle Program Manager and Apollo Flight Director—when I was heading up Flight Software, he was with RSOC [Rockwell Space Operations Company] at the time. We were having lots of quality control problems in Flight Software, and I was struggling with it, and he turned to me and he said, "Do you know what you've got to go do, John?" I said, "What?" He said, "You've got to convince those 300 or so people who work for you, who sit at computer terminals and that's all they do all day, that at the other end of their keystroke is a real solid rocket booster, is a real main engine, and is a real person." He said, "If you can communicate to them that every keystroke they make directly touches real power and real people, you won't have to worry about quality control."

I've realized that Glynn was right. That's what I tried to do with Shuttle Return to Flight, even in the SE&I. I brought the SE&I org [organization] down to go through the hangar and



look at *Columbia*. I made a real point that there had never been an astronaut assigned to the Integration Control Board. I *got* an astronaut assigned, I got astronauts involved. We made a real point of communicating that what you do matters to real people in their real life.

It was easier with X-38. I remember one day with Don [Donald E.] Reed—we had a pyrotechnic device hang up on the X-38, and we had to go safe it. I turned to Don Reed, who was my Flight Test Engineer—he's now head of the flight tests for Orion—and I said, "We need to go safe this by putting this pin in the device." He had done it many times, and I said, "Do you feel comfortable with this?" He said, "Yes." I said, "Do you understand what the configuration of the device is?" He said, "Yes." I said, "Are you ready to go do it?" He said, "Yes." I said, "All right, go do it." Then I kept doing what I was doing; I was holding a meeting. We were dealing with the aftermath of the flight, and he went off to safe, and I said, "And come back and tell me when it's done." He went off and did it and came back. He knew I was concerned about his safety, and therefore he trusted me when I asked him to go do something, that it would work out.

It was easy when it was just us together. It was a lot harder in a big program. I was really surprised when I got into Shuttle—it's not that people didn't care about the crews. It was that all the other noise around them drowned out their ability to respond to that concern. The big programs are noisy. There's so much noise. "What's our contractor evaluation? Who's getting promoted? Who's saying what to whom? What paperwork is being processed? Who's stopping what? What budget issues are working? What schedule are we working?" There's all this noise that drowns out concern for safety of flying. It's fascinating how that happens. And all those are valid concerns. We've got to live within a budget. We've got to live within a schedule, or try to meet a schedule, or use schedule to organize our work so that we can have a disciplined process.

We've got to have processes that we follow. People are going to move up and people are going to move down. People are going to move to the side. All of those things are going to happen, but there is a constant background of noise in the big programs, and people lose sight of that.

Unfortunately, it takes big accidents to bring people's sight and focus back onto that. If we can maintain that focus, people will do the right thing. But the problem is all the other noise drowns out the message. So I think as a manager, you can't just say, "Safety is job one." You can't just say once, "I'm concerned." Every decision you make has got to reflect the concern for the safety of flight, and it's exhausting on a manager. It's exhausting to go through every decision and go, "All right, have we really worked this from a safety of flight perspective?" It exhausts people, and after a while, they get callous to it because they can't deal with the energy of it anymore.

I think that it would be a very reasonable policy to say, "Senior managers need to rotate through programs on a regular basis." Three or four years is about all you can take asking that tough question every day, keeping safety right up in the forefront. Because after three or four years, the noise has reached such a level that you are listening to the noise and not to the other stuff. You've got to comment and deal with so many issues. That doesn't mean that safety is always the way it's got to go. Sometimes we've got to budget cut. We've got to make trades. Sometimes the schedule gets totally hoked up and we have to take some risk with that. But what happens is that pretty soon all sorts of other noise starts getting up in there.

I can't tell you how many times in Return to Flight I was told that the Shuttle Program was going to get canceled unless we flew by the next launch date. We shifted the launch dates almost two years from the initial launch date, and you know something? The Shuttle Program didn't get canceled. The Program actually stood down for another year after first flight, and it

didn't get canceled even though people said. The way programs in NASA get canceled is by lack of attention to safety, not due to lack of meeting all their other goals and noise. You've got to constantly be working the signal to noise ratio, driving down the noise of the other stuff and driving the signal of the safety message up.

One of the things that the guys who are working on Constellation now come and tell me about—one of the things they didn't appreciate about X-38 was how I insulated them from all the noise. It's a manager's job not to take the input from [NASA] Headquarters [Washington, D.C.] and then translate it down to the troops, but to take the input from Headquarters, talk among some key lieutenants, decide what the strategy is, and not let everybody have to deal with it. Budgets, schedule pressures, political pressures. Your manager's got to take that load and not translate it down, and when the managers start translating it down, the noise floor starts coming up and the important signals start getting lost. That's an exhausting job, absolutely exhausting. Some people are very good at it. Arnie [Arnold D.] Aldridge is a good example, excellent at it. Leonard [S.] Nicholson, excellent at it. Other people, it absolutely wears them out.

WRIGHT: What would you consider to be the greatest challenge that you had to overcome working at NASA during those projects?

MURATORE: Tom [Thomas W.] Holloway, very famous Program Manager, head of Flight Directors for a long time, told me something very interesting. He said, "People ask us what our greatest resource is, and we always say 'our people.'" He said, "We have some bright people, but the truth of the matter is we don't have any better people than anybody else has. Our people aren't our greatest resource." He said, "Our sense of mission is our greatest resource. When we

lose our sense of mission, we are in the most jeopardy. When we have a high sense of mission, we can overcome any obstacle." He said, "The program is fueled by young people who come in with stars in their eyes, with dreams of doing great things, and who work with incredible energy. It's further guided by people, much older, more experienced, who work just as hard with the greatest of energies. Where we get in trouble is where we lose the sense of mission. We get wrapped up in politics, we get wrapped up in budget and schedule. We get wrapped up in personal issues." He said, "If we want the best for NASA, we've got to keep our mission, focus on what's our mission." If our activity is not clearly, demonstrably, absolutely, most effectively supporting that mission, we've got to change what we're doing. No matter how painful or how difficult. Because our people sense it, and then we no longer get the best out of them.

When I came into Recon [Reconfiguration Division], the people in Flight Software had lost their sense of mission; it was just a job. They didn't see when they hit the keyboard that that keystroke was going to turn into a command that fired a solid rocket booster. The Mission Control Center: the people were totally worn out and burnt out, and we brought the whole pirate thing in, and we made them believe that we were going to build the world's best control center. We did, and it's delivered ten years of flawless operation, even though the technology jump that we made was so dramatic that people were just terrified of it. In X-38, the people were burnt out at 12 different attempts at a CRV. We built a real CRV, we built a space vehicle. We were within a year of shipping it to the Cape and flying it on the Shuttle when we finally got canceled. In Shuttle Return to Flight, we built a sense of purpose about protecting the lives of the crew and getting the Space Station assembled.

The challenge in all of them is always the same: Are all our activities aligned with our mission? It's amazing how we acquire activities that aren't aligned with our mission and

processes that aren't aligned with our mission. Do our people feel that that sense of mission is overriding everything that we're doing? They come to NASA because of that sense of mission. If they have that sense of mission going, there's nothing we can't do. If it's physically possible, if it's realizable, people at NASA can do it if they believe in it, if they have it that it's their mission. Once we lose that sense of mission, we go into "Keystone Cops."

Finding what the mission was and instilling that sense of mission in our people has always been the difficult challenge, and it's probably the most difficult challenge today because I think people are confused beyond all shadow of a doubt right now about our mission. They understand the importance of Moon and Mars. They don't see how the big, long transition from the end of Shuttle is going to work to the beginning of a new program, they see the new program getting stretched out, they don't understand the science content of the Moon, they don't understand what it builds in terms of capabilities for us. They see downsizing going on all around. They don't understand what the mission is, and that's the biggest challenge. I think the cool thing about it is that's something we can do something about it. It's something totally within our control.

WRIGHT: What would you tell young people that want to come on board that are interested in being a part of the program, of the Space Agency?

MURATORE: A lot of people do seek me out, and I do give a variety of sets of advice based on their individual situations. When I was a young second lieutenant one of the captains taught me was that programs have rhythm. They have times and natural rhythm. They have times for intense work and intense progress, and they have times where it's a lot slower and a lot harder to

see. Right now, the larger, total NASA program, the manned program, is struggling with a time of low energy. Even though we're trying to take on these huge tasks of PDRs [Preliminary Design Review] and finishing up the Station and getting the Shuttle retired, it's low energy in terms of our mission. So you've got to really look deep inside, and it needs to be very carefully thought out.

If you're on a football team in high school or college, and there are cheerleaders and there's the band and there's pep rallies, then there's tons of energy. X-38 was like that. There was tons of energy. Then you may be on the chess club or the debating club, and there probably is not that level of energy and intensity. Coming into NASA right now is a bit like joining the chess club. We may become state champions and there may be cheerleaders and pep rallies about the chess club, but it's going to be a bit of a while before we do that, and there's going to be a lot of work before that happens.

It's a good time to get in, but when I came to NASA—I'd been at NASA less than a month when we flew STS-9. We flew STS-11 two or three months after that. On STS-9 was the first Spacelab. On STS-11 was the first use of the backpacks, the manned maneuvering [unit, MMU]—we were doing new stuff all the time. It was like that during Gemini, it was like that during Apollo, it was like that during Skylab. When I first started, in the Air Force, we were in more of a doldrum period between trying to get STS-1 off, and it was more of a chess club: internally motivated, slow building time.

My advice would be: First of all, don't come to NASA because it's a good career move. It's a horrible career move. It's going to consume your life. Come only if you have a total passion for doing it. Come only if you're prepared to do it, knowing that you're making a big difference but that no one will ever know that you did. Next, come prepared, knowing that it's

going to be chess club for a while. Come prepared to challenge constantly your bosses to not waste your time with activities and processes that are not aligned with our mission. I think that our people have got to challenge our managers and say “no.” Say, "No, this is wasting my time. Why are you having me do this? We're pursuing a design option that we know can't work. Until we fix this problem, it's not going to work."

We're doing this unfortunately in Constellation right now—we know we've got a huge weight problem, we know we've got a big performance problem, yet we continue to move along the baseline and move to design reviews knowing that we've got a huge problem. And everybody knows that the work they're doing is going to have to be redone when we finally solve the huge problem. Other people may debate that with me. I'm not an expert. I'm not in the middle of the Constellation Program. Maybe I've interpreted it wrong, but that's pretty much the way I understand the situation in Constellation. Our people know that, and they're demotivated by it. If we said, "Stop everything. We're going to go deal with this problem, and we're going to fix it, and then we're going to go move on, and we're going to take whatever painful medicine is necessary to fix it," including delays, lack of milestones, stopping, changing contracts, whatever pain is necessary—our people will respond to that a lot more effectively.

I joked around with some of the guys at my retirement dinner, I said, "Now that I'm retired, of course, I'm now eminently qualified to tell you how to succeed at everything that we're not succeeding at." I want to say on the record that the problems are always easier when they're from the outside versus from the inside, so I'm offering a lot of advice that is very hard to implement, and I recognize it's hard to implement.

But people know when what you're having them do doesn't drive the mission forward and when you're wasting their life, and they resent it. Rather we not have them working than we

have them working on something they resent because they know it's not going to get the mission done. My advice to young people would be: come in, be prepared, it's going to take a long time to get this exciting again. It will, but you're going to have to challenge an awful lot to get to that exciting point. It's going to be very unpleasant, because there's nothing more unpleasant than challenging senior people.

WRIGHT: Would you like to share some thoughts about that as well, on your successes and failures at challenging senior people on issues?

MURATORE: Challenging them usually wasn't the problem. Communicating with them effectively usually was. During Mercury, Gemini, Apollo, people worked 24/7. It built a culture of working 24/7, of unbalance in our lives. The whole time I worked at NASA, I led a very unbalanced life. As a result, I think rather than really sitting down and communicating with people, I just walked into a room and started talking loud, talking fast and cursing, and telling everybody what to do. As I look back on it, I could have done that a lot better. Part of it, though, was driven by the culture of imbalance.

I had a great coach. In the middle of Shuttle Return to Flight, I really did some things that really weren't very good. They were done out of the most genuine of good motivations, but they caused some real problems in the program. They brought in a coach for me, Roger Mellott and he taught me an interesting story. He said, "Stand up," and I stood up. He said, "All right, stand on one foot." He said, "Now I'm going to come over, and I'm going to push you over." He kind of pushed on me a little bit, and he said, "How easy is it for me to push you over?" I said, "Well, it's really easy." He said, "Now stand on two feet, spread them apart. Now, I'm going to



push on you. How hard is it for me to push you over?" I said, "It's a lot harder." He said, "John, your entire life, starting as a test conductor working before STS-1, as a flight controller, as a flight director, you've lived your entire life on one foot." He said, "If you're going to succeed, you have to learn to put the other foot down and spread your stance a little bit."

I think if I had my stance spread a little better, instead of getting frustrated, angry, mad, cursing, yelling, screaming—which I did have somewhat of a reputation for doing—about problems, never about people but about problems—I think I could have been a lot more effective. In advice to younger people, or to managers: I'm telling you to have total commitment to the mission, but you've got to find a way to have enough balance so that you don't get knocked over. I think it's possible to be totally committed 100% to the mission and to doing what it takes to succeed in the mission and still have some sense of balance.

I remember Gene [Eugene F.] Kranz talking to us—the first year I was there—about the need for everybody to work one hour a day overtime if we were going to meet the launch manifest. I was like, "Yeah! Yeah, let's go do that! Because we want to meet the launch manifest, because we want to get the Shuttles up, because we want to build a space station, because we want to go to the Moon and Mars, because we want to do good things for people, we want to build solar power stations, and we want to find new drugs," and all the things we were going to go do with the program. The program naturally, due to its Apollo origins, doesn't have a culture of balance. People are unbalanced, the program is off balance. So what this next generation's got to go do is introduce some balance. One of my deep regrets is that sometimes I contributed to the imbalance. Sometimes I contributed to the balance; I guess those are some of the prouder moments. But sometimes I contributed to the imbalance.

WRIGHT: Before we close, I know you brought some notes. I want to give you a chance to take a look at those.

MURATORE: Let's see. There are three keys to solving any problem we have in NASA, and they all come to: build a model, instrument it, and test it. The model can be a computer model, it can be a mathematical model, it can be a physical model. It doesn't matter what it is, all of our engineering problems are solved ultimately by the same mechanism, which is model it, instrument it, and test it. I would really advise anybody who wants to get into managing programs to develop expertise in those three areas. Because in terms of either physical modeling or computer modeling, in terms of what instrumentation can tell you and what it can't, and how to do tests, and what structures tests to give you good information and what doesn't—those are really three key practices that every senior manager at NASA should be an expert in.

I always, as a manager, kept my “hands on” in at least one technical aspect of everything that we were doing. When it was X-38, the joke was I was the third best instrumentation guy on the project. We had two instrumentation guys, and when they got overloaded I would go and wire stuff up, and build stuff up, and test stuff. It's important, no matter how senior you get, to have a technical interest that you use to keep your mind alive and to maintain a different perspective to give you a little bit of a view from the bottom of the program up. That's really important.

Don't be afraid to admit you don't know. Admitting that you don't know something is usually the first step in a process that gets you somewhere, and acting like you know when you don't usually is the key to a process that is a dead end and a stagnation. Wayne [Hale] wrote a very famous memo that says, "Eliminate 'I don't know' from your vocabulary." To me,

recognizing that we all have limits of our knowledge is important. I think you can use it as a crutch—"I don't know," meaning, "I don't know with absolute certainty, with a math model and test data and eight decimal places." No. I'm not saying that everything has to go to that extent. What I'm saying is you've got to be willing to go, "I just don't know. My engineering judgment, my intuition, my available data, test analysis—I just don't know the answer to that question." I think it's really important. We need to be able to go do that.

Money management in NASA—it's really important to know all the details of your financials of everything you're doing, because money controls everything. I started with very small projects, and I knew every expenditure down to the penny. It developed, for me, a sense of being able to go through the financials pretty quickly and understand what was going on, and I had some friends that worked in the financials and they were really good, and they taught me a lot about financial systems. There's a tendency among NASA managers to treat their financials as, "Well, the finance and business people do that, and then they come in once a year and we do POP [Program Operating Plan]." That got the Space Station Program in real trouble, it's gotten the Space Shuttle Program in real trouble, it gets programs in real trouble on a regular basis.

Dealing with the financial world, the business parts of it, are an everyday part of the business just like every other part of the business. I think you need to take the time—as painful as it is—to really learn that and to go through it on a regular basis. You need to not let concern about the finances dominate your thinking because it can become noise in this environment. But the first time you're managing a large program should not be the time when your financial people start talking to you about the difference between committed and obligated. You need to learn that much earlier, and you need to have an expertise about the financials. I think people tend to

avoid the financial and the budget and they view it as a one-year thing, and you've got to view it like your checkbook and you deal with it on a daily basis.

Don't be afraid to ask for help. Tom Holloway taught me an interesting lesson about that. We were dealing with a really tough problem with Shuttle Flight software. I was in charge of Shuttle Flight software, and I got all my guys in a room and I made a decision. I came to Tom Holloway and I briefed him, and Tom hated the decision and he really ripped me up a new one about it. Then afterwards, I said, "You know, Tom, I'm really sorry about that," and he said, "Did you talk to Jon [C.] Harpold about this?" Jon was just a genius and had been with the program a long time—he was Head of Flight Dynamics at the time at MOD. I said, "No." He said, "Maybe you should have gone and talked to Jon Harpold about that."

I don't know if it's my own particular failing or it's a wide failing in NASA, but I think we're a little afraid to ask each other for help. When you go ask somebody for help, it's got to be focused. Everybody's very busy, you've got to do the homework ahead of time, but I think generally when you go ask people for help, if you've done the homework ahead of time so that you don't just drop a problem on there—when you say, "Hey, I'm trying to solve this problem, but I need some advice, I need some expertise," I think they generally respond and go out of their way to help you with it.

I think the biggest thing that has made a difference in the projects I've been involved in where I've been successful—there's this great movie with Kirk Douglas called *Cast a Giant Shadow*, and it's about the founding of the state of Israel. Very appropriate, since today's the 60th anniversary. Evidently, during one of the very first wars of the founding of the state of Israel, they had to build a road to Jerusalem, and in this scene the guy's talking to his engineer and he's saying, "You've got to build me a road in 24 hours that goes from here to here." And the

guy's explaining all the difficulties of how the road is, and Kirk Douglas bends down and grabs a rock, and he says, "I need you to build me a road." He grabs the rock, he picks it up, he moves it out of the way and drops it. He says, "There. I've started it. You go finish it."

One of the toughest things we have in NASA is bending down and picking up the first rock. We keep waiting for it to be the perfect time, the perfect place. "Have we picked the right rock? Have we grabbed the rock in the right way?" My parting piece of advice would be: just start throwing rocks and let it sort itself out, because it does. Things change, each rock you move gives you greater experience about how to make rocks move. Then what you've got to do is be flexible: "I've moved 100 rocks, but the 101st I'm going to move differently because I've learned from moving 100 rocks." The toughest problem I think we really have is being ready to pick up the first rock and move the first rock. Most problems at NASA get solved by starting with an imperfect solution and then improving it based on experience, not with starting with a perfect solution. So if we would get more comfortable with starting with imperfect solutions and then improving them with experience, I think we'd get a lot more done. That's my parting comment.

WRIGHT: We'll let that stand, and I'm sure we could ask you more, but we'll stop for now so we can let you go on to what else you have to do this evening. Thanks so much for coming in and sharing your thoughts.

MURATORE: My pleasure.

[End of interview]