

**NASA Administrator Daniel Goldin
Remarks at the
15th Annual NASA Continual Improvement and Reinvention
Conference
April 27, 2000**

Conference Theme: "proper planning prevents poor performance"

Good morning everyone.

We have had about a dozen times as many successes as failures. But today I want to talk to you about the causes and cures of failure.

We all know that NASA's record of accomplishments has been outstanding. I am proud NASA has saved approximately \$40B from planned budgets for the American taxpayer and is doing more for less. Since 1992, NASA has launched 146 payloads valued at a total of \$18B. Of this number, 136 payloads were successful. Our total losses amounted to 10 payloads costing less than 3% of the total.

And the theme of this conference, "proper planning prevents poor performance," is very appropriate, but it is not the whole story. Planning is the just the beginning -- we have to fully follow through with the plan.

That is apparent, because mixed in with our successes, we have had some pretty serious failures, most notably the Mars Climate Orbiter and the Mars Polar Lander.

However, we are not alone, the Russians have suffered recent failures of their Proton launch vehicle - a vehicle with an excellent long-term record of success. The Japanese have lost 2 H-2 vehicles. ESA's first Ariane 5 launch failed -- and we have lost a Delta and a Titan-4

We are at a strategic inflection point with the cold war warriors and Apollo era veterans retiring. And a revolution is upon us with a technology explosion of digital information systems. In spite of this, we must not repeat the errors of the past. This is "blocking and tackling", not rocket science.

So my message to you today is a demand that both NASA and industry work to continually improve our performance. We will not always be successful, but in everything we do, we need to remember the words of Ben Stein, "The human spirit is never finished when it is defeated. . . . it is finished when it surrenders."

So instead of surrendering to our failures, we need to recommit ourselves to continuous improvement. This will make our successes even more spectacular.

What continuous improvement means to NASA is a commitment to mission safety through leadership and accountability. By this I mean an intolerance for failure by everyone involved: from NASA to our contractors and from junior engineers and scientists to senior managers and executives.

Why have our recent failures occurred? Could they have been prevented?

My 35 plus years of experience and my own "gut feeling" tells me they occurred because we forgot the recipe for success. And yes, the failures could have been prevented. Like with most failures, once we do the forensics we find we could have done a better job.

Again and again, I keep hearing the same old lessons learned. As Yogi Berra, "its d j vu all over again". And as I said, this is "blocking and tackling, not rocket science.

However, being experts at mission forensics is not our goal. Our responsibility -- all of us -- is to do whatever is necessary to prevent failures. We owe that responsibility to the American public and to ourselves.

Success begins leadership and the subsequent accountability. Everyone involved -- NASA, industry and academia -- is personally accountable for the success of our missions. And when there is a failure, we should take it personally -- not to the point that we become discouraged and give up but to the point where we challenge ourselves not to let it happen again.

Accountability means when there is a failure we find out why and we fix it. No excuses or apologies, we fix it.

That's the can-do spirit the NASA-industry team is known for.

Recent failure analysis reports made clear that the problems with the Mars Climate Orbiter, Mars Polar Lander and the shuttle wiring system were all avoidable. And the challenges we face provide many opportunities for improvement. Let me be clear here -- these are opportunities for both NASA and for our contractors.

However, there is a common thread in all of these failures. They are not caused by a single error, but by a sequence of errors that build and propagate due to a lack of leadership and accountability. The missions also went from conceptual design to detailed design and development without the necessary checks and balances to break the failure chain.

Now let me tell you in detail what happened with the Mars Climate orbiter and Mars Polar Lander. I am not singling them out except as well documented, solid object lessons about what can go wrong when we let our guard down.

First mars Climate Orbiter.

What went wrong .

It wasn't the use of British units versus Metric units, it was the FAILURE of the system to catch this very, very basic error.

Other factors contributed to the ultimate demise of MCO

1. The spacecraft design had the aerodynamic center of pressure offset from the spacecraft C.G. which resulted in more desaturations of the momentum wheels than normal. This had a major impact on navigation but the spacecraft navigators were not aware of it.

2. There were software problems very early after launch. JPL could not read LMSC navigation data files for several months. These problems were never communicated upwards. And, there was NO V &V for the entire system.
3. There were small "inconsistencies" noted after each burn that were and not effectively communicated upward to resolve and understand. If this had been done it would have uncovered the real reasons for the inconsistencies . The WRONG units were being used.
4. As soon as MCO was launched, it was be handed over to a separate group responsible for operating all Mars missions. There was no "cradle-to-grave" transfer of key, knowledgeable people into the operations phase and, the operations group was all ready busy working MGS problems in addition to inheriting MCO.

Mars Polar Lander (MPL).

What went wrong

Most probable cause was that the retro engines shut down prematurely due to a spurious flag being set when the landing legs were deployed.

Other factors contributed to the ultimate demise of MPL..

1. There was no software regression testing and no system wide V and V. The "I have landed" flag should have been continuously refreshed until the landing indicated touchdown. However, it was activated by mistake when the landing legs deployed causing the descent engine to shut down several meters above the Martian surface.
2. Test of the landing systems did not reveal a improperly wired sensor. The wiring problem was later found and fixed. But the entire system was not re-tested and, in fact, remained faulty. Basic test and verification practice was not followed.

3. Addition funds were requested before the start of Phase C/D, but denied by Headquarters. JPL signed up to do the mission anyway, lured by the challenge.
4. The group psychology of "can do" led to an attitude that things were tight but OK. Rigorous independent reviews would have exposed that too much risk was being taken.
5. There was NO senior mentor assigned to a "young" project manager to advise on risk and test philosophy and when it was time to RAISE THE RED FLAG!

These are all errors that should have been caught.

How do we eliminate these problems? How do we hold ourselves accountable and fix them? We focus on people, processes, execution and advanced tools and technology. This is what Art Stephenson, the Director of NASA's Marshall Space Center, concluded in his report on the Mars Climate Orbiter . And, I agree.

For the next several minutes I want to give you my views on some of the ways I believe we can fix what is wrong. I want to first talk about what I believe we can do in the near-term -- over the next year or so. Then I will tell you about the future as I see it.

We need to assure that our expectations do not exceed our resources. That we can do what we say we are going to do and that we can do it right the first time. This means we must be sure we have the financial resources, technical resources and human resources. And, that we have the ability to test and verify what we eventually plan to build.

We cannot allow ourselves to over run cost because we "oversold" the program from the beginning. We cannot delay the schedule because we need to develop new technology we did not know we needed. We cannot assume we will find the right people to lead and staff our projects when we need them. And, we cannot skimp on verification and validation testing -- hardware, software and the entire integrated system.

Also, we cannot get too cocky or be content to trust our own knowledge or judgement. We need to build some and test some to assure anomalies and errors are caught early and do not propagate. We must test and simulate the way we fly -- and fly the way we have tested and simulated.

And, something I believe in very strongly -- every project and mission must have rigorous, probing independent technical and programmatic reviews, with highly experienced people to force us to explain our decisions and our actions. And to dig out any issues effecting mission success that we may have passed off as unimportant. Their findings and recommendations must be formally tracked to closure.

And by experienced people I mean a blend of senior people with broad aerospace experience and others with unique ideas and new insight. If you recall it was Richard Feynman who conducted the famous test of a shuttle solid rocket booster O-ring with a beaker of ice water and butterfly clamp.

These should not be "stop and go events". Our independent review teams should be standing oversight panels that follow a project from mission start to end. They are our gremlin hunters who are empowered to stalk the shop floor, look over our shoulders and take us to task when they sense something might be wrong. This is not the traditional 2 days of viewgraph watching.

They are also our own "hired" gremlins -- not part of the development team -- constantly looking for our weaknesses and finding ways to "force a failure". And we should give them the resources to do some homework and a bit of their own independent analysis.

To do the job correctly the first time, we must plan our missions correctly, up front. If we do this and we fail, we have no excuse. And we should offer no excuse. We are accountable, so when we fail -- and it will occur - we find out why and fix it.

We must exercise great rigor and discipline in the formulation of projects. This begins with establishing achievable goals, having strong systems engineering to properly define the requirements, flow

them down and establish a sound understanding of performance, cost, and schedule margins.

Equally important is to clearly understand the minimum success criteria up front to allow for de-scope options and decision points, should problems occur. Constraints such as resources, planetary launch windows, launch vehicle payload capability, and limited landing site data, should not be an excuse to drive the project to unacceptable levels of risk.

It is a critical responsibility of senior managers in NASA and industry to assure that everyone involved in a mission fully understands the purpose of the mission and the specific requirements of the mission. These requirements must be documented and we must assure that any changes are understood thoroughly and their impact quantified before implementation.

This means designers, implementors and operations personnel must be included in the definition and flowdown of requirements from the very beginning. Too often we do detailed planning early in the mission development process and only include operations as we near the launch date. This only adds to mission risk and it should not happen.

When we develop our requirements we must also determine the risk we are taking. Risk is a part of every NASA mission, but risk can be assessed, quantified and managed. Today, risk is based too much on qualitative evaluations and judgement based on "experience." But this only works when we have a strong, relevant legacy to build on and when we have highly experienced engineers and managers to rely on.

As we build and fly more spacecraft deeper into space we will venture more and more into territory we have never explored before. More and more, as the cold war warriors retire, and as the number of missions expands we will rely heavily on less experienced managers and engineers. But, lack of experience is not an acceptable excuse for failure. Adequate mentoring training and institutional support must be mandated now, not as a last minute fix.

Risk can also lie in the inherent nature of the mission itself. When we first proposed to land a spacecraft on the moon, there were some theories that they would sink into a sea of loose dust and disappear.

When we first sent Viking to Mars, we did not know the composition of the atmosphere or the condition of the Martian surface. Both entry and landing were considered risky. And as we consider sending a spacecraft into the ice covered oceans of Europa, we cannot be sure what to expect.

But we must know what the risk is. We have to quantify what we know and bound the uncertainty of what we do not know. Only when we know the risk and can mitigate it to an acceptable level are we ready to commit to a mission.

And risk reduction does not stop when the design begins. Concern for mission safety must be pursued throughout the life of the mission. It should never be assumed and we should never be too confident to question matters of safety.

We know we should do this, but we do not always do it.

Why?

Because it is too easy to "rationalize away" risk. Our assessment of risk is too qualitative -- too heavily based on opinion. This leads to errors in judgement that can destroy a mission. Risk and uncertainty must be dealt more like engineering parameters such as temperature and stress level. We must be able to analyze risk and believe the results.

For the time being, our remedies lie in the tools and methods available to us -- fault tree analysis, failure effects and modes analysis, probabilistic risk assessment, experienced managers and extensive independent review.

Today an increasing challenge to our missions comes in the form of software, but the tools we have for developing and verifying our software are disturbingly inadequate. Nearly 2/3 of all software programs are not delivered to their customers because they don't

work. And only about 2% of all software programs are delivered on time, while meeting project requirements. Even then, there are typically 5 to 10 errors per 1000 lines of code in commercial products.

The code that is delivered is also very expensive. Typically, the production rate for code that would be used in a NASA mission is about 15 lines per day per person. To a large extent this is because the time spent debugging code is about 5 times what it takes to write it down the first time.

Software reuse and COTS are often useful in reducing costs and reducing risk but not always. When we use COTS software, we typically only use about 30% of the capability of the systems, but we potentially inherit all the errors. Maybe we will not encounter them, but we can not be sure.

And with the lives of our astronauts on the line, we **must** be sure.

Take, for example, a GPS software package that was being considered for the shuttle. This was COTS software that had been further verified by another government agency. When NASA did its own V&V, we found the software had an error that would totally lock up the shuttle's main computer.

Or consider the failure of the Ariane 5. Guidance software from the Ariane 4 was used in the Ariane 5. However, the Ariane 5 achieved a higher horizontal velocity than the Ariane 4. This caused a register to overflow, which in turn caused the primary and secondary inertial reference units to shut down, which generated spurious control signals to the engine nozzles, driving the vehicle to a destructive angle of attack. Why did this occur? The software was reused without completely verifying whether it was valid for use on the Ariane 5.

Today our ability to efficiently develop and validate software is woefully inadequate. But our dependence on software is only going to increase in the future.

For comparison: Voyager had only 3000 lines of code. Galileo had 8000 lines, Cassini 32,000 lines, and Mars Pathfinder 160,000 lines. Future spacecraft will have millions of lines of code.

If we cannot verify and validate the code we write -- or the COTS software we buy -- we might as well not bother with the mission at all.

The planning and execution of a thorough, bottom-up verification and validation program is essential. This demonstrates end to end performance margins, provides understanding of the system, trains people, and develops contingency plans.

Having the tools for effective integration and test of ground and flight software is also essential. Integration of the system and emulation of time delays between the ground and flight software must be tested, and should be verified with the ground hardware and space-borne hardware.

Industry and NASA must have the capability to do reliable verification and validation of all critical software functionalities. And there must be the capability for independent V&V as well. NASA is committed to developing and using the IV&V facility we created in Fairmount, West Virginia, for this purpose. This facility will have independent authority to conduct IV&V on appropriate NASA missions, and to certify other organizations to do so as well.

Ultimately, the success or failure of a mission depends on the skill and dedication of the people involved and in the training and leadership we provide them. It is the responsibility of senior managers in NASA and industry, and none of us should be satisfied with the job we are doing. I am definitely not satisfied.

When team members lack necessary skills we must develop them through realistic academic and on-the-job training. If they lack experience, we must build it through mentoring. We must make sure that an adequate balance of experience people and energetic, talented youth -- **they** are our future leaders

Program and project managers must foster a team-building environment. They must instill confidence in everyone involved that they can accomplish the mission together.

This can only exist if there is trust and open communication among all team members. The work environment must be one where people feel free to address issues without fear of reprisal and perhaps even rewarded for reporting problems. No one should be intimidated or afraid to report bad news. And, I mean at every level of the organization.

Team members must also be free to ask for help when they need it. And managers and institutions must be alert to it and ready to provide it -- whether it is technical review, relieving an excessive work load that could lead to mistakes, providing mentoring or guidance from more skilled or experienced co-workers or gaining vital experience through additional training.

The teams and their institutions must be alert to people who are working too hard or are overstressed. Some stress is good. It means people are committed and trying their best. But too much stress will destroy team unity and increase the likelihood of mistakes. Sometimes, it is an indication of inexperience.

When you go back and look at your own organization see if you are doing what you should, with the resources you have today.

For the future we need tools and a work environment we do not have today. We need tools that can simulate an entire mission end-to-end; explore options and contingencies; Tools that allow us to accurately determine risk and cost.

When we conduct a stress analysis and the results indicate we have exceeded allowable limits or a thermal analysis indicates a critical component will overheat, we do not pass judgement on the results and choose to ignore them. Risk and uncertainty must be treated the same way.

We need to depend heavily on learning and knowledge-based tools to capture the experience from prior missions and the insight of

experts. In doing so we will enable systems to learn and reason for themselves and to extract information and knowledge from complex distributed databases. These higher level supervisory design methodology concepts would then be integrated into engineering life cycle analysis and design tools to develop and institutionalize a safety oriented smart design process.

With this knowledge we will be able to design systems with levels of mission safety unachievable today.

These tools and the knowledge we gain will help us create a more effective workforce. We could use case studies as educational tools and let people do mock designs under the supervision of an intelligent agent. No tool will replace smart people, but smart tools can create smarter people.

This is NASA's vision of what we are calling Design for Safety. It is a vision we are making a reality.

This fall NASA will host a workshop to discuss the limitations of today's tools and methods and the necessary technology and roadmaps we all need to develop and implement Design for Safety.

Yes, we have faced some challenges in the past, but the human spirit will not be denied. We will never surrender to failure.

When we work as a team, we will meet those challenges. And when we do, America's future space systems lead the world, and with God's good graces, the market cap of all our aerospace companies will rise like a hot rocket.

Do not shy away from the challenge, and never surrender to failure.

Thank you very much.