How Do You Deal With Anomalies?

I am writing this column on Thursday, July 14, 2005. Yesterday I was one of more than 25,000 people who spent all day at the Kennedy Space Center's Visitor Center with the intent of watching the launch of the Space Shuttle Discovery's STS-114, Return-to-Flight mission. If you've never been to a launch, as I hadn't, it is a *very* long day.


Scott Barber

The scheduled launch time was 3:51 p.m. My wife, my stepson and I had to arrive at the Space Center by 9:30 a.m. After hours of standing in lines, going through security checkpoints, waiting in more lines and being transported to the observation area—virtually the entire time spent outside, unshaded from the 90-degree-plus heat and the central Florida summer sun—we finally arrived at the viewing area to wait the final two and a half hours until launch. Needless to say, I was quite excited. Getting to see a shuttle launch was something I'd wanted to do for 20 years. We had our cameras, a camcorder, binoculars, lawn chairs and everything else we could think of to make it a comfortable and memorable experience.

Of course, if you follow the space program at all, you already know what happened next.

About 10 minutes after we settled in and got our binoculars focused on the shuttle, the announcement came over the loudspeaker: "Ladies and gentlemen, I'm sorry to have to inform you that we just received word that the launch has been scrubbed for today. Please return to your buses."

Needless to say, we were all rather disappointed. At that point there were no answers to the questions of "What happened?"; "Is there a new launch time/date set?"; or "Will we get a refund for our launch tickets?"

By the time we made it back to the Visitor Center, I learned by visiting space.com on my Web-enabled PDA that the reason the launch was scrubbed was a faulty fuel sensor.

Within the hour, the following status message appeared on space.com: "NASA experts acknowledged that the sensor problem—which they described as an intermittent event with no obvious cause—represented a difficult challenge."

The sensors "for some reason did not behave today, and so we're going to have to scrub this launch attempt," launch director Mike Leinbach told the launch team. "So I appreciate all we've been through together, but this one is not going to result in a launch attempt today.

"Launch control said it will take some time to figure out the problem."

By the time I arrived at home, additional information was available:

"The fuel tank contains four sensors that show how much hydrogen remains in the tank. One sensor indicated that the tank was almost empty, even though it had been fully loaded with 535,000 gallons of liquid hydrogen and oxygen.

"A faulty reading could cause the shuttle's main engines to cut off prematurely or to burn for too long, either of which could be potentially disastrous for the craft and crew."

And by the time we'd finished eating dinner:

"Similar fuel-gauge problems cropped up intermittently during a test of Discovery back in April. The external fuel tank, along with cables and electronics equipment aboard Discovery itself that are associated with the fuel gauges, were replaced, and even though NASA could not explain the failure, it thought the problem was resolved and pressed ahead with launch.

Hale defended that decision.

"We became comfortable as a group, as a management team, that this was an acceptable posture to go fly in," he said, "and we also knew that if something were to happen during a launch countdown, we would do this test and we would find it. And guess what? We did the test, we found something and we stopped. We took no risk. We're not flying with this."

Shuttle program manager Bill Parsons stressed that it was not clear whether the problem was with the fuel gauge itself, or with other electronics aboard the spacecraft.

NASA is looking closely at the possibility that flawed transistors in an electronic black box aboard Discovery might be to blame. The box used in the April test also had bad transistors, and when it was removed from the shuttle, the problem disappeared. Managers now suspect a manufacturing defect with these transistors.

Parsons nixed a fueling test of Discovery's replacement tank in June, over the protests of some engineers. Such a test would have pushed the flight later into July, and Parsons and others maintained that the ultimate test would come on launch day. Moreover, Hale said there was no guarantee that the malfunction would

Scott Barber is the CTO at PerfTestPlus Inc. His specialty is context-driven performance testing and analysis for distributed multiuser systems. Contact him at sbarber@perftestplus.com.

have turned up during a tanking test.

The issue came up again at launch readiness reviews earlier in the week, and to everyone's satisfaction, it was deemed an "unexplained anomaly," according to Hale.

The launch scrub cost NASA an estimated $616,000 in fuel and labor costs.

Fascinating, isn't it? Reading that, my feelings suddenly shifted from disappointment to empathy. How many times have I made similar decisions during testing? Think about it. How often have you seen the results of a performance test without a single anomaly? If your experience is anything like mine, the answer is probably "rarely." In fact, when I think back, those times when I did not find any anomalies in the results, my instincts told me to question the validity of the tests.

It immediately occurred to me that I'd have recommended exactly the same approach to handle a performance testing anomaly that the NASA engineers took for the fuel gauge. Try to reproduce it. Explore the results in more detail. Possibly swap out, rebuild or instrument the offending code or machine. Then eventually decide that the project would be better served by proceeding with our testing and "keeping an eye out" for recurrences than by continuing to search for something that may well never happen again.

So the question is, "How much effort should be put into trying to understand a single anomalous test result?" Obviously, it shouldn't just be blindly discounted, but what if it really was an unrepeatable quirk, a testing error or even someone in the server room mistakenly sitting down and trying to access the wrong test server? Ultimately, there needs to be some heuristic for deciding to accept the possibility of recurrence and just move on, since, as Bill Parsons went on to say in his statement to the press, "It's difficult to find a glitch that won't stay glitched."

I started thinking about what my heuristic was for performance test results. My first thought was of an article I wrote several years back that borrowed statistical models from several industries to come up with a point of reference in determining outliers in response-time data. The summary is that it appears to be statistically valid to say that data points that represent less than 1 percent of the entire data set and are at least three standard deviations off the mean are candidates for omission in results analysis if (and only if) identical data points are not found in previous or subsequent tests—or, in layman's terms, "really weird results that you can't immediately explain accounting for a very small subset of the results which are not identical to any results from other tests." I'm still pretty comfortable with that, so let's agree to use that as a working definition of an anomalous result.

Malesuada a, aliquam at, tincidunt at, urna. In imperdiet est. Curabitur interdum enim sed sem

Once I have detected a results anomaly, I realized, I always ask myself the same questions to guide my next steps:

- If this happened one out of every hundred possible times in production (the worst case, based on our definition), what would that mean to the company/product/client/user?
- Would stakeholders consider delaying the project, going over budget, etc., over this worst case?
- Is this worst case more or less severe than other issues I am likely to uncover by continuing testing in other areas versus spending more time on this?

No matter what the answers are, I always document the anomaly somewhere so that it can be found easily, and then I modify my tests to highlight that anomaly if it should crop up again.

Thinking about these questions led me to the realization that all I am really doing is a very simplistic risk analysis that could be restated as "Is the potential cost of a particular failure greater or less than the potential cost of trying to eliminate the possibility of that failure occurring?" I guess that doesn't really surprise me, but the realization that this thought process occurs with virtually every performance test I execute—and further, that it is so easy to explain—makes me wonder why I didn't think of it sooner. I already set the expectation that the priority of the planned performance tests needs to be reviewed to potentially revise testing priorities every time an issue is detected. Should we not review priorities based on anomalies as well? NASA did.

They chose to move on toward launch, keeping an eye on the fuel sensors, and they ultimately caught the problem before disaster could strike. Realistically, that's a pretty positive outcome. From what I can tell, the launch would have been delayed if they had chosen to chase the anomaly when it was first detected, and there was no guarantee that it would have been found before Discovery showed up on the launch pad anyway! At least now they have a known issue and two data points to analyze.

Does that make it worth the $600,000-plus price tag? I can't answer that, but I can say that it's hard to stay disappointed about not getting to see the launch when I step back and realize that in the same situation, I most likely would have done exactly what NASA did.

I guess it's time to review my approach and my reference material to explicitly address how to handle test anomalies. ☒