

Liar, Fool or Both?

Robert N. Charette, ITABHI Corporation

“**D**esignated liar” is the term used by congressional staffers and the national press to describe the military briefer who is called upon to testify before Congress on ailing defense programs. The epistemology of this quaint term stems from a belief that military briefer will assert anything, including outright lies, to justify their program’s existence. This belief is reinforced by past experience, i.e., the almost universal discrepancy between a defense program’s currently projected costs and its later, much higher, actual costs, as well as by the many defense programs canceled for cause that right up until their cancellation, were “on schedule and meeting performance requirements.” Some congressional staffers and members of the press do have some sympathy for these briefing officers, regarding them (and their bosses) as only incompetent fools, instead of skilled liars.

We in the software community should consider co-opting “designated liar” into our own vernacular. After all, the record of software projects meeting their original cost, schedule, or performance criteria aren't so much better than defense projects, and may be worse. And how many projects that never see the light of day are reported to be 90% complete, or are said to be on schedule right up until the morning of the next major milestone, when an 18 month slip is announced? Alternatively, what about those announcements for new software products that never appear, AKA, vaporware?

No Shortage of Guilt. So, who should it be? Who in our community earns the right to the title, “designated liar”? How about project managers? After all, they are responsible for the making plans, developing budget and schedule estimates, controlling the development, and monitoring project progress. Better yet, how about chief designers, since they are the ones who provide much of the information to the project managers in the first place? I know, let’s nominate marketing managers, since they sold the project to the customer in the first place. Any seconds on that nomination?

No matter whom we pick, I am quite sure they would loudly object. “Me a liar?” they would say, “How dare you! I never lie to the customer about our capabilities or our product. What I tell them is only the truth. Perhaps I exaggerate a little, but I never outright lie. That would be wrong ... unethical even. And if there is one thing I am, it’s ethical.”

Funny enough, this is the similar line of argument a military briefer gives. As described by Lt. Col. Mark F. Cancian (USMC) in a recent *Naval Institute Proceedings* article, most military officers see themselves as embracing a higher ethical and moral standard than the civilian community at large, and thus are generally more truthful.¹ The military, on the other hand, view journalists, congressional staffers, and the general public as uninformed regarding the reasons costs and schedules forecasts usually end up being wrong, or why hammers end up costing \$100, and worse, not caring to find out. Military officers see journalists and staffers as pursuing their own political agendas, without regard for the truth.

Three Broad Questions. As Cancian points out, the crux of the problem lay in the perception (and one’s perspective) of truth, and shows there is a very thin reed separating truth, interpreted truth, and a lie. What a military briefer may see as the “truth,” a congressional staffer or reporter may view from their perspective as at best the “incomplete truth,” if not a clear-cut lie. Cancian continues to point out

that ignoring this “truth-lie” perception gap inevitably leads to a total loss of credibility and trust for all involved, leading to unfortunate consequences for the military and nation as a whole.

To demonstrate why, Cancian poses several pointed questions concerning lying for military leaders to consider as a means to enhance awareness of the problem. These I believe are very apropos to the computing community as well. For I believe the credibility and trust of the computing community is eroding quickly with an already technology skeptical public, which we ignore at our peril. If we do not act soon, we will end up commanding about as much respect as a military briefer. So, let’s consider three of Cancian’s questions:

How optimistic can an estimate be before it becomes, in effect, a lie? To try to answer this first question, we need to define what is a lie. For our purposes, a lie is “an intentionally deceptive message which is stated.”² A lie, therefore, involves a deliberate choice to deceive—whereas, an erroneous statement based upon good faith efforts may be deceptive, but it would not be considered alone a lie.

Looking at the data on software projects histories, which indicate only about a 40% average on-time, within-cost rate, an outside, neutral observer might question the veracity, competence, or at the very least, the perception of reality, of those making software estimates.³ How could so many project estimates be so far off? The pattern looks very suspicious, like underestimating a project’s cost and schedule is pervasive and intentional in the software business.

Those making software estimates would argue that while some estimates are inevitably low balled, in the main, the estimates are based upon the best available

¹ M.F. Cancian, “Are You Telling the Truth?,” *Naval Institute Proceedings*, Dec., 1994, pp. 37-41.

² S. Bok, *Lying: Moral Choice in Public and Private Life*, Vintage Books, New York, 1989, p.13. The message can be written, verbal or non-verbal form.

³ Exactly what are “the numbers” for software projects being on-time, in-budget, and meeting original performance specifications is open to debate. For various and often conflicting data, see R.

information. Estimating software projects is an inexact science—you are being asked to predict from uncertain data a result that may happen years in the future. This uncertainty often increases dramatically because the software projects are frequently unprecedented. Furthermore, even some of the best estimating tools, many of which are mandated by the customer to use, only advertise an accuracy of being within 20% of the eventual actual results 70% of the time.⁴ Also, what does optimistic mean? Being aggressive? In addition, what about estimating too high a cost on a project? Wouldn't that be lying too, or even criminally fraudulent if done on a government contract?

Our observer might point out that estimating tools assumes some level of realism in determining their initial input. Customers make the same assumption. It is hard to believe that the pressures to succeed don't color the estimator's input to be more optimistic than realistic—that price-to-win estimating isn't dominant. How many estimators use “below average experience” as the input parameter for their cost and schedule estimates as compared to the projects that overrun, slip, or fail because of “lack of experienced personnel”?

One could continue, with point and counterpoint, but we need to recognize that to an outside observer, our record of success, or lack thereof, in meeting our cost, schedule and performance estimates, appears suspiciously intentional. We may argue, like the military briefer, that outsiders just don't know the difficulties involved. However, it is unimportant—the perception does. Just as you don't care how much it costs to produce a car—you are only interested in the price. The rapidly expanded trend of outsourcing computing and the resulting loss of jobs can be at least indirectly linked to this perception of information technology costs gone

Garner, “Management Meltdown,” *Open Computing*, Jan. 1995, pp.36-43, and C. Wong, “Are You a Failure,” *Open Computing*, Mar. 1995, p.18.

⁴ B.W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ, 1981. The accuracy of estimating tools is also debatable. This, in combination with reference 3, adds to the suspicion.

amuck.⁵ Although we cannot guarantee the certainty of our estimates, it does not obviate us of the duty to state clearly, unambiguously and often, the (un)certainly our estimates possess.

How much “spin” can you put on a story before it becomes a lie? Spin is a political term meaning to put a favorable light on a situation, especially a disagreeable one, to control the public perception of it. Putting the best face on an unpleasant situation is not wrong per se but as Cancian asks, can “spin” go to far?

A reason this question is important to us is that the gap between our explanations regarding what has gone wrong, and what actually did go wrong, significantly influences public perception. If the gap is perceived to be a chasm, all explanations that are given will be seen merely as ruses to deceive, with credibility suffering a result.

Take the case of the design error in the Pentium chip. Intel claimed that the error was unimportant to most users, with Intel’s chairman Andy Grove saying, “You can always contrive situations that force this error. In other words, if you know where a meteor will land, you can go there and get hit.”⁶ Intel claimed that average user would likely encounter the error only once in 27,000 years of spreadsheet use, and refused to replace the chips (although the error was being corrected in newly manufactured Pentiums). Others such as IBM, however, claimed the real chance for error was much higher—a business with 500 Pentium PCs could have up to 20 errors a day in recalculating spreadsheets just 15 minutes daily. IBM chose to discontinue marketing the chip. As the public outcry grew louder, Intel relented somewhat and announced it would replace only those chips proved as being used for intensive scientific mathematical calculations. This caused even a larger outcry, finally forcing Intel to agree to replace any defective chip no questions asked.

⁵ R. N. Charette, “Getting Rid of Your IT,” *Software Management*, Apr. 1994, pp. 12-14.

⁶ B. Ziegler and D. Clark, “Computer Giants’ War Over Flaw in Pentium Jolts the PC Industry,” *Wall Street Journal*, 13 Dec 1994.

Did Intel's spin go too far in this instance? In other words, did Intel intentionally mislead? From Intel's perspective, I doubt they believe they did then, or now, since it admitted the error when the flaw was discovered. Nevertheless, Intel was also economical with the truth, as it kept quiet about independently discovering the flaw months earlier. Although Intel was probably right in its assumption that the chip flaw was of little consequence to most of its customers, Intel had, in effect, reserved the right of making that judgment to itself without their customers' consent. Although Intel has seen few returns of its defective chips and no real loss in business, its credibility decreased. Although not generally seen as a liar, the episode has made Intel seem rather foolish and hard hearted.

In April 1993, the core component of the FAA's Advanced Automation System (AAS), the Initial Sector Suite System, passed its first agency-monitored test. At the time, the acting FAA administrator said, "We have a system that I believe we have our arms around, and we have absolute confidence in our schedule."⁷ Similar positive statements were made by the prime contractor to the same effect. The administrator's statement was made a month after a 14-month slip of the AAS program was announced, which had already suffered an earlier 19-month slip in 1990.

In December 1993, the FAA declared the AAS project would cost an additional \$1.2b, but insisted the schedule was still okay. However, by April 1994, it was clear that the program would have to suffer yet again a 9 to 30 month slip, cost billions of dollars more, as well as require a major restructuring.

What happened here? How did the program go from having the "absolute confidence" of the FAA administrator in making its schedule to being a disaster in a span of 8 months? How does a program go from being on schedule to as much as 30

⁷ J. M. Smith, "Air Traffic Control Core System Gets Through First Test," *Government Computing News*, 26 Apr 1993.

months behind three months later? Were the upbeat and positive statements by the FAA a spin too far? On the other hand, were they just lies no one was expected to believe anyway? Several congressional representatives thought they were being told the truth, claiming that they had been consistently misled on the status of the AAS.

As in the Pentium situation, truthfulness got lost in the truth. True, starting December 1993 the AAS was still on schedule, but only because the paper on which the schedule resided hadn't been changed. The costs and schedule were changing so often that our neutral observer would have to assume that any schedule was incorrect. Our observer would have a difficult time not concluding that some of the statements made were more than merely foolish, given that there were:

- numerous assessments of the program costs and schedules by GAO and others that stated they were significantly underestimated, and
- that much of the delay was caused by FAA-initiated changes.

Are there situations where literal truth is not expected? Some may be feeling, “What’s the big deal, here? Everyone knows that estimates are lies, and that we bury our disasters under cover stories. It’s a game where everyone knows the rules.” Many customers of software in fact do view estimates of cost, schedule, or performance as only partial truths from the beginning, and consider them as bids in a game of poker. Deception is part and parcel of the game, and what is stated by a vendor is appropriately discounted. Further, if producing and submitting estimates are part of a bargaining or negotiation session, there is wide (but not unlimited) legal latitude given to the truthfulness of the estimates.⁸

However, what if one party is unaware of the game? For instance, most commercial software licenses have “as is” clauses similar to this one:

⁸ G. R. Shell, “When Is It Legal to Lie in Negotiations?” *Sloan Management Review*, Spring 1991, pp. 93-101.

“In no event will X be liable to you for damages, including any lost profits, lost savings or other incidental or consequential damages arising out of your use of or inability to use the software, even if X ... has been advised of the possibility of such damages.”⁹

Experienced software users expect that software comes with bugs, and work around them as best one can. We have been trained over the years to be patient, and to give developers the benefit of the doubt, since we too are not entirely without sin. Nevertheless, our industry, and our future jobs are now dependent on neophyte users, who have a much different perspective of a consumer product. They look at “as is” clauses as little more than bright neon signs declaring, “Whatever we claim about our product, it’s not necessarily the truth.” To say the least, this does not inspire much trust in computing, or those involved in it.

For years, we have been trying to get computing into the mainstream. Well, we succeeded, and have for an industry that affects so many, been pretty unfettered by governmental interference. The industry has been also held up for respect and emulation. But, we better realize that our success may be sowing the seeds of our own destruction if we don’t start to see that the basic nature of our industry, our interface to the public, and our responsibilities to that public, have radically changed.

Unless we begin to understand how our actions as an industry are being perceived by the outside community—realize that to many we are looking more and more like liars and or fools—and begin to do something to keep our credibility from eroding further—expect someone else, i.e., the government, to “help” do it for us. High profile and often incorrect reporting of events like Denver’s baggage handling system fiasco to flaming of new users to the Internet only add to the erosion of our credibility. Five years ago an urgent call for the IEEE to act on ethical standards

⁹ J. E. Rigdon, “Frequent Glitches in New Software Bug Users,” *Wall Street Journal*, 18 Jan 95.

was made that seems to have been ignored.¹⁰ We had better act now before it is JTL—just too late.

As has been said about Microsoft knowing how to compete, but maybe not know how to behave that may soon be the whole industry's epitaph. I don't know about you, but I would hate to be perceived as equivalent in trustworthiness to a military briefer.

*A version of this article first appeared in IEEE Software in July 1995.
Copyright © 1995 ITABHI Corporation. All rights reserved.*

¹⁰ M. C. McFarland, "Urgency of Ethical Standards Intensifies in Computer Community," *IEEE Computer*, Mar. 1990, pp. 77-81.