BUILDING BETTER GOVERNMENTS

Many thanks to Neville Holmes for introducing the discussion about the science behind building a better form of modern governments (The Profession, “The Design of Government,” May 2009, pp. 104, 102-103).

However, I disagree with the idea that bifurcation can make things better even with proper education.

Concerning bifurcation and education, in the US we have many well-educated persons in the CIA, FBI, NSA, and so on who collect and analyze information, and we have the FRB, NEC, FMOC, Treasury, SEC, and so on for operations. Yet we failed to prevent this global economic crisis.

Is it more difficult to collect and analyze information about the crazy practices of loan companies, various banks, and insurance giants than it is to gather information about terrorists?

In addition to bifurcation and education, it’s important to encourage people in the system to do good things and, at the same time, discourage them from remaining ignorant. This is called responsibility. It could be enforced by establishing practical and up-to-date rewards or penalties and demanding compensation for wrongdoing. This should be started right now instead of just putting more taxpayers’ money in the economy to try to recover from previous wrongdoing. Or we really will need to worry about the fate of our profession.

Jun Wang
jun.wang@ieee.org

Neville Holmes responds:

But systems of rewards and penalties bring their own problems. For example, it should be obvious that jails are quite dysfunctional as a penalty for wrongdoing. Rightdoing is a consequence of healthy sociality, and this comes about through success in education. Jails should focus on education rather than incarceration, and the computing industry and profession could greatly help make such—indeed all—education socially successful. For starters, see The Profession, “Supporting Acquisition of Basic Skills,” Mar. 2008, pp. 104, 102-103.

CHANGING THE HEALTHCARE SYSTEM

David Grier’s column in Computer’s June issue (The Known World, “Marking the Fall of Sparrows,” pp. 9-12) was truly compelling.

I agree, as most will, that creating a workflow will change our healthcare system. We know this by experience, as we also know that it will cost more, take longer, and do less than we expected. We also know, from (painful) experience—the IRS system, SAP in the government, and countless other monstrous projects—that our record has a lot to do with the human condition. It’s in our nature to never be quite satisfied with our undertakings. We begin every project with a set of requirements that we will, inevitably, change until the project is completed or canceled because part of the human condition is to unceasingly want more. Maybe it starts in childhood, when we are trying to please our parents.

If our history in computerization and software is an indicator of future performance, perhaps instead of throwing our resources wildly at this massive endeavor and fooling ourselves once more that great leaders and copious amounts of money can overcome anything, we should accept that this task is more complex and pervasive than any we have attempted and therefore take small, manageable bites.

There are likely, as Grier pointed out, many competing interests that no amount of planning, no spreadsheets, no Gantt charts can predict.

And above all, we cannot hope to outmaneuver our own ambition and intellect. With this in mind, maybe, just maybe, like the Moveable Feast, we should plan to win the war by seeking small victories.

Mike Maitland
mike.maitland@TransCore.com

ETHICS ISSUES

The feature articles in Computer’s June 2009 issue raise some interesting points, yet miss some key ones that are related.

Tetsuo Tamai’s stock exchange example implies the question, What can be learned?

My response is that we need a lot more than ethics to stop such problems. When software people do follow the various ethical codes that have been promulgated, they often are overridden by management that is in a hurry and wants to cut costs.

At a minimum, we need licensing of software engineers along the line of other engineering professions. These SWEs would be registered by the states as is done with other engineers as well as with other professions such as CPAs, doctors, and nurses.

Maybe we need to license and register managers too. Then we might avoid the concomitant problems of their allowing idiots to needlessly download unencrypted databases onto laptops, complete with SSNs, that are frequently stolen. Until management is forced into protecting the public by encrypting all databases, limiting access by need to know, and taking other operational measures to
ensure that problems cannot occur, identity theft as well as other fraudulent schemes will continue to be perpetrated with ease.

It is a fundamental principle that you need to label every item with a unique sequence number and the current time. If that had been done in the stock market example in Tamai’s article, the database problems could have been avoided. This is rarely done, based on red-herring strawman arguments about speed, storage size, cost, development time, and so on.

Related to that stock exchange problem is the article discussing software security. In both cases, a proper systems architecture is not mentioned. Without a systems architecture, you end up with the endless cycle of patches that we now have. Far better to architect a secure system that does not need patches at all.

With a proper systems architecture followed by a proper systems design and systems implementation, of which software is merely a supporting component, it would have been routine to design a stock exchange system without such logic land mines, as well as to build absolutely secure systems that are immune to scumware and other security problems.

While acting ethically is good, ethics are not a solution to our problems. By themselves, ethics give a false sense of security. We need to address the bigger issues, and then ethics would happen as a side effect.

William Adams
williamadams@ieee.org

AUTOFLIGHT SYSTEMS

In “The Public Is the Priority” (June 2009, pp. 66-73), Donald Gotterbarn and Keith W. Miller omit one important ethical principle favored by those of us who analyze incidents: Refrain from making imposing public statements on technical matters about which you know little.

The authors illustrate well the reasons for this principle through their Case 2. They introduce the 2008 Qantas accident and suggest that “a decision to give instant control to the plane’s flight control system when the autopilot shut off because of computer system failures” … “was not in the best interest of the public.”

Autoflight systems have been doing exactly this since they were invented over half a century ago, and no pilot or engineer I know would have it otherwise.

The ATSB preliminary analysis hints rather at an obscure bug with the Flight Control Primary Computer, as well as a yet-undiagnosed fault in one of the air data subsystems. Let us hope that our colleagues at the companies concerned can discover what and how and devise remedies.

We can indeed hold moral views arising from this and other incidents, such as that critical software and interfaces need to be rigorously proven free from every possible source of error. However, most software engineers would agree that best practice is still some way from that ideal, and back when flight control systems were cables and pulleys, we were not close to it either.

Concerning the Aeroflot upset, I feel strongly that children should not be placed at the controls of commercial passenger jets in flight, and that it is silly to suggest that the system design should accommodate such an eventuality.

Peter Bernard Ladkin
ladkin@rvs.uni-bielefeld.de

The authors respond:

Our description of the Qantas accident was overly simplistic. Peter Ladkin and other experts we have consulted agree that a problem in the Flight Control Primary Computer (FCPC) seems to be involved, in conjunction with anomalous “spiking” in one of three Air Data Inertial Reference Units (ADIRUs). There were reports of ADIRUs spiking in different airplanes earlier, but without the diving behavior of the Qantas incident.

The issue of data integrity in avionics systems is complex. These systems include multiple techniques to deal with possible false data readings, including the possibility of human pilot overrides and algorithms that attempt to distinguish between anomalies from errant sensors and actual emergency situations. Through a complicated series of events, at least one of these algorithms yielded a “false positive” identification of a dangerous stall-inducing climb that was “corrected” when the FCPC ordered a steep dive. This occurred twice during the Qantas flight in question.

Interested readers can find the interim report from the Australian Transportation Safety Board at www.atsb.gov.au/publications/investigation_reports/2008/airr/pdf/AO2008070_interim.pdf. At this writing, the Board has not issued its final report.

We contend that complex system interactions like this create ethical as well as technical challenges for all involved. This case, no matter how badly Peter Ladkin thinks we described it, deserves further study and public discussion. Even when bugs are obscure, life-critical software decisions are ethically charged for software engineers and for the people their software affects. We hope that the larger theme is clear in the article.

Editor’s Note: For the May 2009 article titled “Challenges Supporting Federated Assurance” (pp. 42-49), Hiroki Itoh’s affiliation should have been listed as NTT (not NTT Advanced Technology).

We welcome your letters. Send them to letters@computer.org. Letters are subject to editing for style, clarity, and length.