DOI:10.1145/2556647.2556655

Article development led by acmqueue
queue.acm.org

"Not invented here" syndrome is not unique to the IT world.

BY POUL-HENNING KAMP

Center Wheel for Success

WHEN I FIRST read the claim that HealthCare.gov, the website initiated by the Affordable Care Act, had cost $500 million to create,[4] I did not believe the number. There is no way to make a website cost that much. But the actual number seems not to be an order-of-magnitude lower, and as I understand the reports, the website does not have much to show for the high cost in term of performance, features, or quality in general.

This is hardly a unique experience in the IT world. In fact, it seems more the rule than the exception.

Here in Denmark we are in no way immune: POLSAG, a new case-management system for the Danish police force, was scrapped after running up a tab of $100 million and having nothing usable to show for it. We are quick to dismiss these types of failures as politicians asking for the wrong systems and incompetent and/or greedy companies being happy to oblige. While that may be part of the explanation, it is hardly sufficient.

The traditional response from the IT world is that the Next Big Thing will fix this, where the Next Big Thing has been a seemingly infinite sequence of concepts such as high-level languages, structured programming, relational databases, SQL, fourth-generation languages, object-oriented programming, agile methodologies, and so on ad nauseam. I think it is fair to say none of these technologies has made any significant difference in the success/failure ratio of IT projects. Clearly they allow us to make much bigger projects, but the actual success/failure rate seems to be pretty much the same.

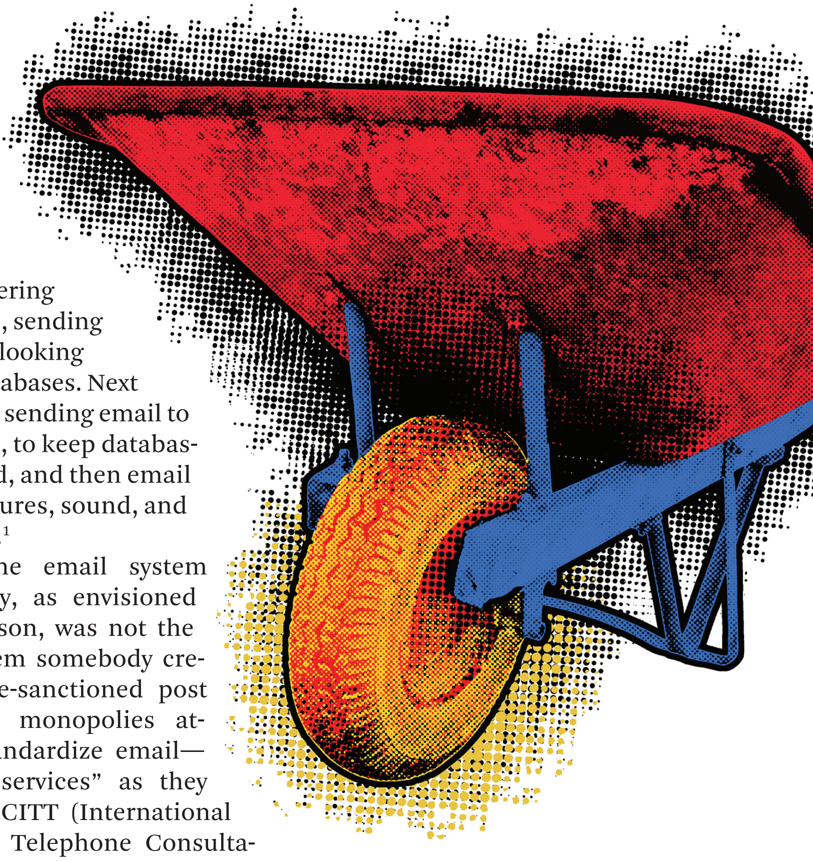
At the same time, there are all these amazing success stories, where a couple of college kids change the way we think about information retrieval with their Google information-scoring algorithm, or a bunch of friends change the way we communicate with their Twitter information-distribution system.

Why, despite politicians' lofty speeches, does that never happen in government IT applications? There is clearly something we are missing here, something we are doing wrong, without even thinking about it. That particular mistake is far more common than it should be in a (so-called) "knowledge economy."

Lessons from Wheelbarrows

Growing up in the countryside, I spent a good portion of my youth operating a wheelbarrow. The European wheelbarrow is a rationalization of the handbarrow, which was basically two planks, two feet apart, with boards nailed or tied between them. One person grabs the two planks at the front, one in each hand, another grabs them at the back, and then they trudge away with their load.

Sometime back in the low thousands, a productivity consultant must have pointed out that if you replaced the person in front with a wheel, then you could get twice as many wheelbarrows moving with the same number of workers. (This industrial application of technology undoubtedly earned the consultant a hefty fee.)



And that is it! That is the very same contraption I lugged around as a kid and the same one I used just a few hours ago for gardening. As anybody knows, using a wheelbarrow is easier than carrying things, but it is still quite heavy work. You lift roughly half the load yourself, you provide the energy for motion, and you must steer it in the right direction, which is difficult on account of the first two expenditures of energy.

While a vast improvement over the handbarrow, the wheelbarrow is stupidly inefficient, at least compared with the Chinese version.[2] Somebody in China was smarter than the Medieval European downsizer and moved the wheel to the middle of the wheelbarrow, so that the entire weight of the load is carried by the wheel. The Chinese wheelbarrow will readily transport two or three times the load of a European wheelbarrow, with the operator hardly breaking a sweat, just pushing and steering, with barely any lifting.

From a management perspective, the Chinese wheelbarrow is identical to the European one: one wheel, two handles, one operator. Looking at it that way, however, we blind ourselves to how differently they work, and we miss the full productivity improvement of the wheel.

In Europe we have known about the Chinese wheelbarrow since at least 1797,[2] yet, to this day, we still sweat while lifting half the load carried on our nonoptimized wheelbarrows.

The "not invented here" syndrome is not unique to the IT world.

I am beginning to think the reason our big IT projects sink is that we make the same kind of mistake: mindlessly replacing human labor with technology instead of solving the actual problem.

Many human jobs can be replaced directly with computers. Email replaced the old telegraph system, delivering the exact same conceptual service: delivering a text message quickly while using hardly any manpower. But delivering text messages was the least email could do—once we got to know it better.

First there were programs answering email messages, sending source code, or looking up things in databases. Next came programs sending email to other programs, to keep databases synchronized, and then email containing pictures, sound, and vice presidents.[1]

However, the email system we know today, as envisioned by Ray Tomlinson, was not the only such system somebody created. The state-sanctioned post and telegraph monopolies attempted to standardize email—or "telematic services" as they called it—in CCITT (International Telegraph and Telephone Consultative Committee) recommendations X.400-X.599,[3] as part of the grand vision of "The Intelligent Network."

They started approximately 15 years before Tomlinson. They spent uncountable millions of all sorts of currencies. They had legislators mandating their way be the one and only legal way forward. And they failed utterly, miserably, and definitively.

Why is it that in IT one person can often do what thousands cannot?

It is tempting to speculate that HealthCare.gov would have worked much better had they given the task to a 10-person company rather than a conglomerate with 69,000 employees all over the globe. I am sure that is a necessary part of the solution, but again, it is hardly a sufficient condition for success.

For one thing, while there are "only" 380,000 words in the Affordable Care Act (also known as Obamacare), the regulations floating from the law amount to 12 million words (and counting). No 10-person company would even be able to read all that verbiage before the delivery deadline had whooshed past.

Interestingly, *The New York Times* reports that HealthCare.gov contains an estimated 500 million lines of code.[4] That is no more likely to be true than the $500 million price tag.

I looked at one of the actual laws that make up Obamacare, the Patient Protection and Affordable Care Act (PPACA),[5] and since I was not going to read all 906 pages, I started in the middle, on page 403. After a few pages I ran into this definition of *patient decision aid*:

"(1) PATIENT DECISION AID—The term 'patient decision aid' means an educational tool that helps patients, caregivers, or authorized representatives understand and communicate their beliefs and preferences related to their treatment options, and to decide with their healthcare provider what treatments are best for them based on their treatment options, scientific evidence, circumstances, beliefs, and preferences."

Reading on, I found the requirements:

"(2) REQUIREMENTS FOR PATIENT DECISION AIDS—Patient decision aids developed and produced pursuant to a grant or contract under paragraph (1):

"(A) shall be designed to engage patients, caregivers, and authorized representatives in informed decision making with healthcare providers;

ILLUSTRATION BY IWONA USAKIEWICZ/ANDRIJ BORYS ASSOCIATES

"(B) shall present up-to-date clinical evidence about the risks and benefits of treatment options in a form and manner that is age-appropriate and can be adapted for patients, caregivers, and authorized representatives from a variety of cultural and educational backgrounds to reflect the varying needs of consumers and diverse levels of health literacy;

"(C) shall, where appropriate, explain why there is a lack of evidence to support one treatment option over another; and

"(D) shall address healthcare decisions across the age span, including those affecting vulnerable populations including children."

Unless Congress thinks of teachers as "educational tools," I think we can take it as written here that they expect this to be some kind of computer program. But read it again and pay attention to the language. When was the last time you saw a computer program that "engaged," "explained," or "addressed decisions?" Or, for that matter, when have you seen a program that "adapted for [...] a variety of cultural and educational backgrounds to reflect the varying needs of consumers and diverse levels of health literacy"?

These paragraphs legislate that Obamacare will fund research in heavy-duty state-of-the-art artificial intelligence—I somehow doubt that is what Congress intended it to say. I posit that Congress worried about having enough doctors and nurses for this new healthcare, so they wanted to use computers to cut down the talking and explaining. In other words, they want to save manpower—by replacing the front man on the handbarrow with a wheel.

I have used a handbarrow once, in an emergency. My fellow campers and I constructed it from two young pine trees, wrapping the sail from our tent around them. Compared to a wheelbarrow, it was both easier and faster, because the front man did not get stuck in any holes or hit any rocks, and he helped with all of navigation, lifting, locomotion, and steering. When we met the first responders, they gently lifted our friend with his injured leg from our makeshift version to their professional handbarrow and carried him the rest of the way to their ambulance on a high-tech aluminum stretcher.

I am absolutely sure that Congress would never replace the front man on an ambulance stretcher with a wheel to save manpower—yet, in a way, they did just that. I do not claim to know the correct way to optimize a healthcare consultation with computers—there may be one, but more importantly, there may not.

Blindly deciding that IT be substituted for humans is unenlightened. IT is not a magic potion that makes unpleasant or inconvenient things disappear. The right thing to do is to ask, as a Chinese engineer did 2,000 years ago, "If we're going to put a wheel on this thing, where is the best place to put it?"

And to realize that two questions were asked.

Blindly deciding that IT be substituted for humans is unenlightened. IT is not a magic potion that makes unpleasant or inconvenient things disappear.

Related articles on queue.acm.org

More Encryption Is Not the Solution
Poul-Henning Kamp
http://queue.acm.org/detail.cfm?id=2508864

Better Health Care Through Technology
Mache Creeger
http://queue.acm.org/detail.cfm?id=1180186

A Requirements Primer
George W. Beeler and Dana Gardner
http://queue.acm.org/detail.cfm?id=1160447

References

1. Borenstein, M. and Linimon, M. The extension of MIME content-types to a new medium. RFC 1437, 1993; http://www.rfc-editor.org/rfc/rfc1437.txt.
2. De Decker, K. How to downsize a transport network: The Chinese wheelbarrow. *Low-tech Magazine*; http://www.lowtechmagazine.com/2011/12/the-chinese-wheelbarrow.html.
3. International Telecommunication Union. ITU-T recommendations; http://www.itu.int/ITU-T/recommendations/index.aspx?ser=X.
4. LaFraniere, S., Austen, I. and Pear, R. Contractors see weeks of work on health site. *The New York Times* (Oct. 20, 2013); http://www.nytimes.com/2013/10/21/us/insurance-site-seen-needing-weeks-to-fix.html.
5. Patient Protection and Affordable Care Act. 2010; http://www.gpo.gov/fdsys/pkg/PLAW-111publ148/content-detail.html.

Poul-Henning Kamp (phk@FreeBSD.org) is one of the primary developers of the FreeBSD operating system. He is widely unknown for his MD5-based password scrambler, which protects the passwords on Cisco routers, Juniper routers, and Linux and BSD systems. Today, he is an independent contractor.