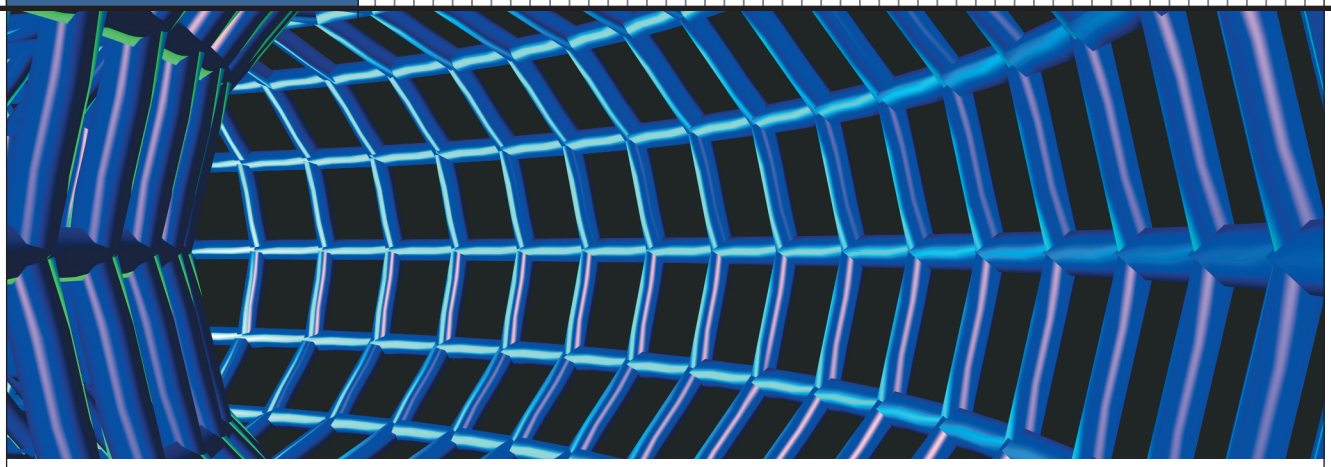

Why the FBI Can't Build a Case Management System

Jerome W. Israel

A review of the problems that haunted the FBI's Virtual Case File and Sentinel case management programs and an examination of the technical reasons for these failures provide the basis for recommendations to help avoid their repetition.

Epic software engineering failures are nothing new in the federal government, but for the FBI, they're particularly bitter.

From about 2001 to 2004, the bureau spent $171 million on the failed Virtual Case File, and it has struggled for nearly eight years after VCF to deliver Sentinel, a $451 million program for which the chances for success and fulfilling the original requirements grow bleaker by the day. The plan was to automate the paper processes administered by the FBI's special agents. Instead, their hopes have been crushed twice, forcing them to continue managing cases and evidence in ways that haven't improved dramatically since the days of J. Edgar Hoover.

How does something like this happen to the world's greatest law enforcement agency? What can possibly be done? Much of what went wrong isn't new, the usual fare that Frederick Brooks documented decades ago in *The Mythical Man-Month: Essays on Software Engineering* (Addison-Wesley, 1995). But apart from that, the FBI encountered complex technical issues on Sentinel, several of which had remained unresolved after VCF. We were so busy tending to cost and schedule that we failed to reckon with these issues. In addition, the project was simply too large, too complex, and too unruly for an organization with minimal engineering expertise.

Unfortunately, the FBI isn't alone. Many federal agencies undertake large IT projects and frequently end up in trouble. This cannot continue. In an era of soaring budget deficits, government must build IT differently.

EARLY WARNINGS

In November 2004, just as we were driving the last nails into VCF's coffin, FBI Director Robert Mueller invited Chief Information Officer Zalmai (Zal) Azmi, Program Management Executive Dean Hall, and me to a meeting that also included Deputy Director John Pistole. The subject: a follow-on to VCF. My heart beat in my throat throughout most of the meeting because of fresh memories from my experience at the National Security Agency, where we worked doggedly on programs designed to convert a technology base focused on the Soviet Union into capabilities that could manage the massive volumes and variety of commu-

0018-9162/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

nications emanating from the Internet. These were NSA's proud turn-of-the-century transformation programs, but they all failed, including our flagship program Trailblazer, which cost more than $1 billion.[1]

Based on that dismal experience, it was no stretch to conclude the following:

- because of their enormous complexity, large IT programs rarely succeed;
- large government contractors, "IT cartels" as former Obama administration CIO Vivek Kundra called them,[2] are usually only out for money; and,
- program managers have a tendency to put a rosy spin on projects even when they're clearly in trouble.

Moreover, the FBI's engineering strength was weak—we didn't have the engineering talent to pull off a major project like Sentinel.

Mitigating our risk by demonstrating some of the key technologies first would have given us confidence as we tackled building production versions of these new capabilities.

But the FBI wouldn't be denied. We moved forward aggressively early in 2005 to deliver a request for proposal (RFP) by the end of the summer. But before the contract was even signed, harbingers of doom should have deterred us. What Alan Wade, the CIA's CIO, said when he addressed senior- and midlevel FBI IT managers in April 2005 was stunning in light of the path we were on. The CIA was through with large IT programs, he told us. They just didn't work. The agency had experienced countless setbacks on massive IT projects. The CIA's new approach, he said, was to break them into smaller pieces, build momentum, and achieve incremental success. In hindsight, we should have paid closer attention.

We also should have built prototypes. We faced many mind-bending business and technical problems—"wicked" problems as technology executives described them in regular meetings with the director. It seemed prudent to mitigate our risk by demonstrating some of the key technologies first. This would have given us confidence as we tackled building production versions of these new capabilities.

In November 2005, the CIO's external board of advisors strongly recommended that we build prototypes for Sentinel. But under pressure to move forward, we didn't take the time or allocate the money to engage in this important early step.

When the FBI sent the RFP to industry in August, we received only two proposals. We should have seen a red flag then. Both proposals were generic and lacked creativity; they definitely weren't equal to the effort the FBI had put into the RFP. With numerous public lessons learned on VCF, we were surprised that neither proposal contained a coherent plan that addressed the FBI's most challenging technical problems.

The FBI's evaluators were split on whether to move forward with a contract award. We were disappointed that so few companies had bid on Sentinel and guessed that many were concerned about getting bogged down on the project and then landing on the front pages of *The Washington Post*. In the end, we moved forward and selected Lockheed Martin.

THREE WICKED PROBLEMS

Several wicked problems haunted us during Sentinel, and they will continue to haunt any case management system development until they're finally dealt with.

Ocean and islands

The "ocean-and-islands" problem was undoubtedly the worst. The FBI operates in a vast ocean of data, which agents can freely access in the legacy mainframe Automated Case Support (ACS) system to advance their investigations. But in this data ocean are islands, which are restricted to only a few. For example, a counterterrorism case might contain data that agents can freely access and view. But if the case agent were to upload newly acquired healthcare information on a US citizen involved in the case, access to the case would suddenly be severely limited because the Health Insurance Portability and Accountability Act (HIPAA) contains strong provisions protecting the security and privacy of healthcare records. Hospitals and doctors must protect the privacy of these records, and the FBI must do so as well. The counterterrorism case now becomes an island.

There are vast numbers of islands in FBI case management, hundreds of archipelagos. The FBI is required to restrict access to grand jury proceedings and individual IRS tax records. Also, if an agent were to upload a transcript, or any other student records, the case would become an island because the information is protected under the Family Educational Rights and Privacy Act.

Nobody knew how many islands there were. Every time a new access control was needed, the Natural programmers wrote one in ACS. Some in the mainframe unit claimed there were more than 20,000 such controls. This became a huge technology issue because we planned to use commercial software in Sentinel, but in 2005, we weren't aware of a commercial solution that offered such fine-grained access control. At one point, we consulted NSA's database research experts about Oracle Label Security, a database security program that could protect records at the row and column level. We were hopeful about OLS, but the research experts

believed that it wasn't "expressive" enough to handle the ocean-and-islands problem.

The access control issue was also a business problem. If we were going to succeed, the operations organizations (the divisions specializing in counterterrorism, counterintelligence, crime, and so on) had to reduce the number of access controls. This was difficult for the FBI because in 2001, Robert Hanssen, a career FBI employee who had access to some of the bureau's most sensitive secrets, was arrested after 22 years of spying for the Soviets. This rocked the bureau and was a factor in VCF's failure.

In February 2005, in a statement to the US Senate Appropriations Subcommittee on Commerce, Justice, State, and the Judiciary, Department of Justice (DoJ) Inspector General Glenn Fine noted that in the Trilogy project, the third phase of which was VCF, there "… was a lack of firm understanding of the design requirements both by the FBI and the contractors. … Additionally, a need for broadened security requirements due to vulnerabilities identified in the Hanssen espionage case affected Trilogy's development." (www.justice.gov/oig/testimony/0502/final.pdf)

The FBI tightened access after the Hanssen episode, but these shifting security requirements hurt VCF. They also inflicted grave damage on Sentinel. We never performed the necessary business process reengineering of access controls to reduce their number, and we couldn't identify a commercial technology that could potentially handle this multifaceted problem.

Migration

The second wicked problem was migration. This was a two-sided issue: migrating off ACS and all of its interconnected applications, and migrating legacy data from ACS into Sentinel. ACS was the 800-pound gorilla in FBI case management. More than 40 applications interfaced with ACS, and in some cases, with each other. For example, the Bank Robbery Statistical Application interfaced with ACS. Applications received data from ACS and sometimes sent data to this behemoth.

During VCF, developers failed to create a transition plan for migrating off the legacy system. They planned a risky "flash cutover." Moreover, nobody knew the number of interfaces among the dozens of applications and how those interfaces were defined. Over the years, the mainframe unit hadn't properly documented them. Because this had been an issue for VCF, we did our homework for Sentinel and identified and defined each interface. We developed methodologies for slowly migrating off the mainframe. Lockheed enjoyed a modicum of success in building new interfaces to ACS, but we couldn't sever the umbilical cord to the legacy mainframe system.

Likewise, data migration was a major headache. The data was dirty, as we should have expected based on earlier migration attempts during VCF. The data had endless problems such as names, addresses, and phone numbers not matching the format specified in Sentinel's database. Migration promised to be very costly and time-consuming.

When the CIO executives discussed the problem in August 2008, we debated one proposition—not to migrate the data at all. This would require agents to work in two databases—ACS and Sentinel—but in five years, so the argument went, the number of agents using ACS would be minimal. Besides, we had already determined that we couldn't shut ACS down completely because a large amount of personnel and case information was stored on tape. If a Freedom of Information Act request came in, and the requested data was on tape, we would have to use the applications on the mainframe to retrieve the data.

At the end of our discussions, however, we decided that we couldn't expect agents to work with two different systems. So we continued down the data migration path, although two years later, we gave up and decided not to migrate the legacy data. Then, in December 2011, the FBI changed its mind again and decided to migrate cases specified by agents.

Migration was a two-sided issue: migrating off ACS and all of its interconnected applications, and migrating legacy data from ACS into Sentinel.

Commercial software integration

For Sentinel, we vowed to use commercial technology because VCF was a messy custom-coded effort. We worked with Lockheed to choose "best-of-breed" commercial products, which we planned to integrate. We developed this approach during a time when industry was extolling the benefits of service-oriented architecture. The promise of SOA technologies and approaches was that applications could be somewhat effortlessly "plug and play."

But some on our team suspected early on that the integration of disparate commercial products was going to be wicked hard. A document and records management application and a workflow product, which included the capability to create forms, constituted Sentinel's core. A midlevel agent once said that the FBI is driven by forms—it has a form for everything. By some estimates, when Sentinel started, the FBI used 817 forms. One of the most common was the FD-302, the interview form. Through business process engineering, we reduced the number of forms required to 122, an impressive effort, even though the Sentinel contract only called for the development of 22.

In Sentinel, an agent was to fill out an electronic form and send it through an approval process (the workflow). Once a supervisor had approved the document, it would be

sent to the document and records management repository. This sounds straightforward enough, but product integration was a nightmare. We bought a .NET workflow product, which by all standards was excellent software, but the core of Sentinel was a Sun Solaris J2EE architecture—integrating the products was nearly impossible. We discarded the $1.6 million workflow product and used less capable software built into the document management application.

Then, the ocean-and-islands security had to be draped over the document management application and workflow software. Yet, heading into fall 2008, we didn't have a viable security architecture. Sentinel was scheduled to run until June 2010, but the clock was now becoming a factor.

Some in the FBI were convinced that strong program management would lead to successful outcomes on Sentinel and any other IT projects in the bureau.

We needed time for many technical tasks, including calling our aging public-key infrastructure into service, a capability that had been used infrequently and only for e-mail. PKI was critical for user authentication at login and for digital signatures. Records had to be digitally signed, or they wouldn't be considered authentic and couldn't be used in court. Without digital signatures, we couldn't certify our records management system, which meant that the FBI would have to continue to rely on paper records, a practice employed diligently over our 100-year history. As a matter of fact, most field offices devote enormous amounts of space to the storage of paper records. And in Washington, D.C., the FBI maintains mountains of paper records in a huge warehouse. The promise of a paperless world was alluring, but it wasn't to be.

PROGRAM MANAGEMENT ISSUES

To offset the FBI's weakness in engineering, we planned to emphasize program management for Sentinel. Any blips in cost and schedule would ring alarm bells in the Kremlin. The conventional wisdom, which still holds today, is that government agencies don't require strong engineering—that is what contractors are for.

2007-2008

The emphasis on strong program management came from the top. In April 2007, the White House's Office of Management and Budget announced the Federal Acquisition Certification for Program and Project Managers (www.fai.gov/pdfs/FAC-PPM%20Excutive%20Summary%20final.pdf). According to OMB, major acquisitions must have a program or project manager (PM) certified at the senior level unless the appropriate agency official granted a waiver.

The knowledge and skills sought in a senior-level PM are "managing and evaluating moderate-to-high risk programs or projects that require significant agency acquisition investment and agency knowledge and experience; ability to manage and evaluate a program and create an environment for program success; ability to manage and evaluate the requirements development process …; expert ability to use, manage, and evaluate management processes, including performance-based management techniques; and expert ability to manage and evaluate the use of earned value management (EVM) as it relates to acquisition investments." This is an impressive list, but it is interesting to note that engineering skill isn't mentioned.

Based on lessons learned from VCF, recommendations and mandates from OMB, and trends in government, some in the FBI were convinced that strong program management would lead to successful outcomes on Sentinel and any other IT projects in the bureau. We required potential PMs to obtain a program management certification. Many attended an eight-week course or a nine-day boot camp and then passed the Program Management Professional exam. The CIO regularly counted the number of new PMPs and reported that metric up the chain. When the FBI's Investment Management Board approved a new IT project, a PMP was assigned immediately. An engineer (usually a computer engineer or computer scientist) was assigned to the PMP.

How we managed IT programs was a subject of constant debate, which centered on the following question: How could an engineer who spent four to five years in college earning a difficult degree be assigned to the number-two spot on a project, next to a person who only had an eight-week certification? Some of the FBI's PMPs were engineers, and they worked out fine. Most, however, weren't. They held an MBA or some other nontechnical degree, which meant they had relatively little engineering understanding.

But we bought into the strong program management approach, and it appeared to be vindicated when Lockheed delivered Phase 1 of Sentinel in mid-June 2007, about 18 months after the contract was signed. Some called it "lipstick on a pig" because during Phase 1, Lockheed installed an enterprise service bus and wrote interfaces that allowed agents to interact with most of ACS's functionality through a Web browser. It was expensive lipstick, about $60 million worth, but many agents liked it. This successful delivery didn't spawn any all-night celebrations, but senior bureau management collectively sighed in relief. Sentinel's PMs, however, were very excited and relished moving on to the hardest phase, in which an independent (of the mainframe) case management system would be delivered.

However, we made a critical error after Phase 1, and it unfortunately metastasized. The FBI had waited nearly 18

months for Phase 1, and senior management wanted future capabilities more quickly. After the deployment of Phase 1 in mid-2007, the FBI gave Lockheed permission to spend three months developing a plan that would provide agents capabilities incrementally, about every three to six months.

At first blush, this seemed like a prudent course. The problem with the path Lockheed was on was that the "Phase 2 miracle"—an independent case management system—was slated to be delivered 18 months down the road. Management thought this was too long. What if the miracle didn't occur? What if Lockheed couldn't pull it off? By requiring the company to deliver incrementally, so the logic went, we could detect earlier whether Lockheed was encountering problems and find ways to mitigate them. We could also deliver more capability to agents sooner. This was an appealing benefit.

During the three-month hiatus, Lockheed also replanned Phase 2. In the engineering change proposal, the new Phase 2 would deliver only a sliver of the capabilities originally specified—management of administrative cases only. These were a small subset of the hundreds of different FBI cases, and they were the most benign.

Administrative cases as a rule didn't contain islands (HIPAA, student records, and so on) that were common in other cases. So Lockheed, with the approval of the FBI's Program Management Office (PMO), planned to focus on a much less demanding set of cases in Phase 2. In addition, in the new Phase 2, only administrative case data was to be migrated, not criminal, counterterrorism, or counterintelligence case data. As a result of this change, Lockheed's total costs increased from $305 to $335 million, and Sentinel's total cost increased from $425 to $451 million.

2008-2009

Throughout 2008, Lockheed worked on the first 10 Phase 2 increments, only about 10 percent of which related to case management or provided any noticeable user benefit. One increment was "improved patch management," another was "improved system monitoring tools." Because of the replan, Sentinel's end date was extended from December 2009 to June 2010 (later it was extended to October 2010), but we were starting to run out of time. Many of the bureau's engineers saw the handwriting on the wall. They instinctively understood that we were "bow-waving" Sentinel's hardest problems to the end of the program, a sure recipe for disaster.

Here's an example of the bow-waving that contributed to the project's demise. Although Phase 1 was successfully delivered, Lockheed skipped, with the PMO's concurrence, the following requirement: "migrate legacy case data to a test database." This task would have provided insight into how dirty the data was. But it was postponed to Phase 2 where, in Increment 7, Lockheed was to "prepare for the migration of administrative case data."

Exactly what "prepare for migration" meant wasn't clear, but evidently, the preparation wasn't sufficient because Increment 19 called for the "migration of the administrative case data," and Increment 24, which was to run through the summer of 2009, specified "transform and migrate production administrative case data into the Sentinel repository." What should have occurred in a single increment had stretched into four. And more migration increments would be necessary in Phases 3 and 4 for the remainder of the data in ACS, the bulk of the FBI's case files.

The ocean-and-islands problem dragged on in a similar way. One of my responsibilities was to chair a critical design review board for all FBI IT projects. In October 2008, the CIO asked me to spend six months on a tour at the Office of the Director of National Intelligence (ODNI). Before I left, six new Sentinel increments came before the critical design review board.

Many of the bureau's engineers instinctively understood that we were "bow-waving" Sentinel's hardest problems to the end of the program, a sure recipe for disaster.

Increment 11—"Implement access controls"—was by far the most interesting. This capability was defined as providing data access controls based on FBI policies. It appeared that we were finally going to confront the ocean-and-islands problem. The PMO presented additional details, which indicated that they were going to build a "rules engine," software that would apply "labels" to case information ingested by Sentinel. The rules or "policy" engine wasn't quite so simple, but essentially, HIPAA data would be labeled one way, IRS tax records another, and so on. When an agent desired to retrieve information, a filter would compare the agent's credentials and attributes to the label and make a decision about whether to grant access or not.

By the end of February 2009, about a month before my tour at the ODNI ended, the Sentinel PMO submitted another batch of increments for a critical design review. Increment 20 stated, "Build and deploy the functional access control policy engine." This sounded like a repeat of Increment 11.

Sentinel's presenter explained what Lockheed had delivered in Increment 11: "identity management and role-based access control." The rules engine had also been delivered, but it only contained two rules. After the review, it was clear that being so far behind on this critical security feature didn't portend well for the future.

Meanwhile, at the Director's bimonthly Sentinel meetings, the PMs arrived armed with executive charts and graphs to use in delivering briefings to the FBI's senior-

most managers, most of whom just wanted the bottom line. At the top of the first chart was a horizontal thermometer, which expressed the project's overall status in red, yellow, or green. From meeting to meeting, the temperature never changed—it was always yellow, trending toward green.

2009-2010

In October 2009, Lockheed missed its deadline for delivering the administrative case management system. All of the problems—ocean and islands, migration, commercial software integration—came crashing down on the project. Even seemingly simple tasks, like producing forms, became mind-numbingly difficult. We didn't like the in-house forms software, so we bought another package. That software turned out to be deficient, so we wrote Java Server Pages. We also replaced Sentinels' servers, large mainframe-like Sun Fire E25Ks, which used substantially more power than we had estimated. After spending millions on the E25Ks, which were nowhere near end of life, we replaced them with smaller, greener servers.

Around September 2010, the FBI decided to complete Sentinel in-house using FBI employees and agile development methodologies.

In October, Sentinel was beset with hundreds of software bugs, including many priority ones and twos. The DoJ inspector general, who had written critical reports throughout the Sentinel program, noted the following in his March 2010 audit: "To resolve concerns the FBI had about the performance of Sentinel, the FBI contracted with an independent team to review Sentinel's software code and related documentation. The team concluded that Lockheed Martin had significantly deviated from accepted systems engineering practices, didn't follow its own published documentation requirements, and hadn't adequately followed testing procedures. According to the team's report, these deficiencies resulted in over 10,000 inefficiencies in Sentinel's software code" (www.justice.gov/oig/reports/FBI/a1022.pdf).

Around September 2010, the FBI decided to remove Lockheed and complete Sentinel in-house using FBI employees and agile development methodologies. The bureau is now hoping to deliver the system sometime in summer 2012.

WHAT'S TO BE DONE?

At least weekly, it's possible to open a newspaper or trade magazine and find an article on another government IT project's woes. The Department of Homeland Security's Secure Border Initiative Network (SBInet) took a beating in the press for years until, mercifully, DHS Secretary Janet Napolitano announced she was shutting it down in January 2011. Before that, the IRS was the whipping horse for chronic IT problems. A colleague once told me that when he held a senior job at the IRS, the requirements process was abysmal, and contractors ran the show.

History is still repeating itself. In 2010, OMB had 26 projects on its high-risk watch list. They earned this dubious honor by incurring significant cost increases and schedule delays, failing to meet mission objectives, frequently revising the baseline, or lacking clear executive sponsorship or leadership. These projects span 15 departments and are estimated to cost $30 billion to complete (www.ombwatch.org/node/11237).

And if $30 billion isn't jarring enough, consider this: OMB's 2010 "25 Point Implementation Plan to Reform Federal Information Technology Management" states the following: "Information technology should enable government to better serve the American people. But despite spending more than $600 billion on information technology over the past decade, the Federal Government has achieved little of the productivity improvements that private industry has realized from IT" (www.cio.gov/documents/25-Point-Implementation-Plan-to-Reform-Federal%20IT.pdf).

Fortunately, OMB's report and a similar one issued a month earlier by the Department of Defense (http://dcmo.defense.gov/documents/OSD%2013744-10%20-%20804%20Report%20to%20Congress%20.pdf) are revolutionary in their recommendations and conclusions. They're critical, almost condemning, of large IT acquisitions and advocate shorter, modular approaches to achieve success and keep up with technology. The government is finally learning from a decade of mistakes, but it will take a herculean effort to kick the large-program bad habit.

As these reports suggest, Congress and government agencies must find new ways to fund IT, and large programs must become rare. What these reports don't point out, however, is the following challenges, which need serious consideration.

Engineering

While large programs usually implode under the weight of their sheer complexity, in the FBI, we didn't possess the engineering strength to stand a chance. If Congress appropriates, say, $40 million a year for an IT program, no agency can capably dispose of that funding if it lacks engineering vision, experience, and skill. This raises the question of how strong engineering is in each federal agency. The US Office of Personnel Management (OPM) must find out before more money flows down the drain, but it cannot do so easily given the way personnel are currently classified. According to OPM, in

2010, the third largest white-collar occupation was the Information Technology Management series, Occupational Code 2210; the number of Information Technology Specialists (ITS)/2210s was 77,814 (www.opm.gov/feddata/html/20wh0010.asp).

The 2210 series covers everything IT. The ITS field comprises individuals with two- and four-year degrees or no degree at all. It lumps together PhD computer scientists and individuals with computer information systems (CIS) degrees, two very different educational backgrounds. In the fine print, the ITS field covers system designers and developers. True design and development work are higher-level technical skills that require degrees in engineering or computer science. The ITS field shouldn't include personnel with engineering or computer science degrees, and words such as "design" and "develop" should be stricken from the job description altogether.

At the FBI, we had to create position descriptions to sort out computer scientists, computer engineers, and everyone else so that we knew exactly what we were working with. From then on, we hired new employees into the correct occupational groups: computer scientist (1550) and computer engineer (0854).

The 1550 computer scientist series definition is well written; the 0854 computer engineer definition needs substantial work. These occupational groups are seldom used, which reflects the bottom line: agencies don't know their engineering strength, much less why engineering is critical for the mission's success.

Measuring engineering capabilities and maturity

Assuming the first recommendation is accepted, what is the next step? How can government agencies assure Congress that taxpayers' IT investments will be successful?

Knowing engineering strength alone isn't sufficient. Agencies must also determine their engineering capabilities and maturity. Why should Congress send $40 million to a government agency when it is only mature enough to manage $10 million? Why should an agency receive funding for a large application development when it is populated with networkers?

If the DoD rates software development companies according to Capability Maturity Model Integration (CMMI), why not apply a similar model to government agencies? Of course, the model should cover more than software development, just as CMMI has evolved into process improvement, acquisition, and services. OPM, with the assistance of organizations like Carnegie Mellon's Software Engineering Institute, should develop a model that portrays the engineering capabilities and the maturity of federal agencies. That model will most likely be a hybrid of many of the extant capability maturity frameworks.

Is this asking too much? In the FBI, management plans the smallest details for special agents, who carry toy guns on their hips at the FBI Academy in Quantico, Virginia, to get used to their new appendage. As the bureau's tremendous law enforcement success reflects, the processes for recruiting, training, and developing agents are very mature. The FBI would rate off the charts in law enforcement capabilities and maturity. Why not devote the same level of attention to government engineering?

Program management

A friend of mine who worked in a smaller government agency ran a software project that was his boss's top priority. The contractors on this project had customized a commercial product by adding tens of thousands of lines of code (bad idea). The senior PM had all the skills OMB demands, but he was a linguist by training and over his head on a software development project. He admitted that he didn't know what software development standards and processes the company had used, nor how the code was tested, the results of those tests, what kinds of bugs emerged, or how fast the company had closed them. He had to hire a software engineer to build his boss's confidence in the code before it was deployed.

Prototyping helps IT personnel work through complex technology, engineering, and business problems to ensure a new capability is technologically feasible before rushing into production.

This isn't an isolated case. The government has numerous nontechnical or not-technical-enough PMs, individuals who have reached the highest levels in government IT program management.

For complex application development and sprawling network deployments, government should staff the PM position with engineers who've had real-world experience building these systems and have transitioned to program management later in their careers. Not all engineers are cut out to be PMs, like some of the deep introverts who live like hermits in the bowels of NSA. But for others, this is a great career progression—work as an engineer in the first part of a career to gain experience and transition later to project and program management to assume more responsibility.

Risk reduction prototypes

We don't perform enough prototyping on government IT projects. Prototyping helps IT personnel work through complex technology, engineering, and business problems to ensure a new capability is technologically feasible before rushing into production. Prototyping also ties in users,

providing an early glimpse of capability to ensure the developer is on the right track.

In a rush to deliver and save money, most view prototyping as an annoying distraction. This is foolish, as Frederick Brooks pointed out more than 35 years ago in "Plan to Throw One Away," an essay that is a plea to software engineers to build a throwaway first. "The management question," Brooks said, "isn't whether to build a pilot system and throw it away. You will do that."

We knew most of Sentinel's top risks from the beginning—the three wicked problems. If we had built prototypes for them, the outcome would have been different. If we had built prototypes, we wouldn't have frittered away millions on software we never used and Sentinel's EVM reporting would have been more credible. Government is particularly poor at cost estimation, which leads to unreliable EVM.

FINAL THOUGHTS

Would stronger government engineering improve IT project outcomes? Consider what happened at NSA. The agency always hires large numbers of engineers and offers them internships, fascinating work, and a mentoring and career program. So why did Trailblazer fail with such a strong engineering program? A major factor was that leadership back then incorrectly believed that NSA's work force didn't possess the skills to tackle modern Internet technologies. Hence, we outsourced our future to industry.

When Lt. General Keith Alexander became the NSA's director in 2005, he indicated to *The Baltimore Sun* that he would take "smaller steps, more rapidly done, rather than try to take one big jump and make it all the way across."[3]

Following this change, the computer geniuses at NSA broke Trailblazer's requirements into "chewable chunks" and solved many of the problems that had stymied Trailblazer. Some of these capabilities were deployed to various theaters of operation, and FBI agents used them. The feedback from these agents was incredibly positive, causing them to privately wonder why the bureau didn't have capabilities like that.

As for the FBI, its best hope is to slowly move significant programs to the Criminal Justice Information Systems Division in West Virginia. CJIS runs the nation's fingerprints databases and the National Crime Information Center used by every law enforcement entity. An engineering feeder college is located nearby at the University of West Virginia. CJIS will buck this idea because it believes it is responsible only to state and local law enforcement, not the rest of the FBI. But CJIS is a pocket of quality IT engineering in the FBI. The "big" FBI needs CJIS. The organization should gradually be expanded.

Criminals and enemies of the state, whether they're terrorists, spies, bank robbers, hackers, or serial killers, have the power of the Internet at their disposal. They use Google to plan their brand of evil, and visit chat rooms to spawn the next World Trade Center disaster. Regardless of the IT direction the bureau takes, it must keep up. Management must regroup, rethink, and move forward because the nation can ill afford an FBI that is technologically weak. C

References

1. S. Gorman, "System Error," *The Baltimore Sun*, 29 Jan. 2006; http://articles.baltimoresun.com/2006-01-29/news/0601280286_1_intelligence-experts-11-intelligence-trailblazer.
2. V. Kundra, "Tight Budget? Look to the 'Cloud,'" *The New York Times*, 30 Aug. 2011; www.nytimes.com/2011/08/31/opinion/tight-budget-look-to-the-cloud.html?_r=1.
3. "Interview with NSA Director Lt. Gen. Keith B. Alexander," *The Baltimore Sun*, 22 Aug. 2005; www.baltimoresun.com/bal-alexanderqa0821,0,7915684.story.

Jerome Israel *was a technical director in the National Security Agency, the FBI's chief technology officer, and, later, chief information officer in the Department of Homeland Security's Office of Intelligence & Analysis. This article contains his own opinions and not the FBI's. Israel received a BA in Slavic languages and literature from Indiana University and an MS in strategic intelligence from the National Intelligence University. He is currently retired in Florida. Contact him at vseznayushe@gmail.com.*