FierceGovernmentIT

Published on FierceGovernmentIT (http://www.fiercegovernmentit.com)

Q&A: Jack Israel on FBI Sentinel and federal IT development shortcomings

July 1, 2012 | By David Perera

Jack Israel

The FBI's many attempts to build a modernized case management system to replace its antiquated Automated Case Support system are the stuff of notoriety in federal information technology, with the first attempt (called Virtual Case File) collapsing in 2005 and a second, Sentinel, coming close to the edge. *FierceGovernmentIT* recently spoke with former FBI Chief Technology Officer Jack Israel about Sentinel and what it says about large federal IT projects in general. Israel wrote about his experiences for IEEE's *Computing Now*; you can read his article in full, here (.pdf).

Jack Israel: I've been in IT development in government for over 10 years. It started at NSA, then 5 years at the FBI, and I finished about a year at DHS. I grew very frustrated working on large IT programs. Because, by and large, I came to believe that these programs just don't work. It doesn't matter who you are, because unless you can logically break them down into very small pieces – I think that's the way to go – they normally fail. They're too big, they're too complex, and too complicated. So one of the things I'm trying to point out in this article is how these large IT projects fail.

But I want to emphasize that I don't want to single out the FBI or pick on them too much. You could go to any federal agency and find examples of large IT programs that have failed. I cited in the article the Trailblazer program at NSA. But you could take the Southwest Border Initiative at DHS, which they finally had to shut down. You could take the early IRS attempts to build new tax systems. What I tried to do is to analyze this specific project, Sentinel, because it's fresh in my mind, and I lived through it. There are some things in this program that were technically unique to the FBI. But, in general, what I'm trying to say is that I think these large IT programs are a huge waste of money, and the bane of the federal government's IT existence.

FierceGovernmentIT: Can you give an overview of what your involvement with Sentinel was?

Israel: I was a chief technology officer at the FBI, which means that I was one of four senior executives under the chief information officer, Zal Azmi. All four of us had a role in different areas in guiding the program. My role primarily was in the area of engineering and providing a board that conducted critical design reviews.

Whenever Sentinel was ready with design documentation, they would have their own internal preliminary design reviews, and then they would come to us for a critical design review, and we'd have the opportunity to see what direction they were going in and whether or not the paperwork was up to snuff, and whether or not we thought they had thought through the design well enough.

We had a pretty good window into the program through the design review process.

FGIT: The actual coding and requirements were supervised by the office of the CIO?

Israel: The initial requirements – they were done under another division, another senior executive whose organization was responsible for enterprise architecture development. That division basically came up with the initial requirements document for Sentinel that went out for bid. Some of my engineers played a role in formulating those requirements because they had experience on the previous program, the Virtual Case File. My division played in a role in requirements development, but I did not actually manage the process.

FGIT: And the day-to-day supervision of [prime contractor] Lockheed Martin?

Israel: We had a division that did program management. They had dozens of program managers. They supplied a program manager for the Sentinel program. That person came out of that division, which was run by another senior executive.

We all had a role to play in Sentinel, just a little bit different angles.

FGIT: Just to set the stage, maybe you can briefly describe what Sentinel is supposed to achieve that ACS currently can't do?

Israel: ACS – the Automated Case Support system – is based on old technology. It's based on an IBM mainframe with legacy database and programming technology, and I would say one of the main things that strikes you as a user of ACS is that you're dealing with the old IBM green screens. You're not dealing with a web-based environment, which everyone is used to from the Internet.

Not only is the interface archaic, but the way that you search data, the way you input data, all of those are archaic. For example, the electronic case file containing all FBI cases is voluminous. In order to search it, the computer rifles through every bit, from one end of the case file to the other. That's just the way the search works. There's no such thing as indexing. There's no Google index of the case file.

What Sentinel was designed to do is come up with automated workflows for all the investigative processes that went on in the FBI. It was supposed to come up with a set of electronic forms for agents to use. It was supposed to come up with new search paradigms, ones that we're more used to today, like Google searches. It was supposed to promote information sharing, being able to share with the rest of the law enforcement and intelligence community information that the FBI had developed.

And, it was supposed to go from a paper-based storage of information, of records, to electronic records management. That may sound kind of strange, because the case file system in ACS is electronic. It's a computer. But, the records themselves that the FBI uses to go to court are, today, still paper. Unless you have a way to certify electronic records, like with digital signatures, you can't depend on those records. They could be altered, they could be tampered with. We were supposed to come up with a records management system in the new Sentinel. This was very important.

FGIT: When you say that the search goes from one end to the next, could you explain that?

Israel: The agent puts in a term to search, but before something pops out, ACS would search from one end of the database all the way to the other. It would just go through all the bits looking for character matches. It was a brute-force search.

FGIT: It sounds slow.

Israel: It is slow. It takes a long time. What they developed to complement the case file was – this is going to sound crazy – the FBI developed what they called a "universal index." We called it UNI for short.

What happened is if you look at a paper case file nowadays, or cases from the past, you'd see interviews, for example, with names, addresses, and phone numbers circled with red ink. And what that means is that the agent wanted a secretary to take those items and put them in the universal index. So instead of generating a Google-like index, what the FBI does is manually build an index of all the people, places and things; you search the index first, and then the index would refer you to a specific case or cases.

So search is very laborious, inexact in the ACS system. One of the things that Sentinel was supposed to do was to dramatically improve search. It was suppose to use Google-like tools for searching, and it was supposed to build a *new* universal index, a large relational database where you can perform more precise searches.

FGIT: So what happened?

Israel: Specifically, there were several areas which were technically very difficult.

We had an idea at the beginning that some areas were going to be hard, that is, wicked hard. One of the hardest things that we never came to grips with is what I call the "ocean and islands" problem.

Basically, speaking in technical terms, you need very fine-grained access control on information in the database. What that means is that if an agent puts information in the database that's HIPAA related, or if you have student records, or grand jury proceedings, or IRS tax records – all those records (islands) have to be individually protected. And, in some cases, they're protected down to a very small number of individuals who can see them. It's incredibly difficult to build an access control environment that can handle all those different nuances. That was one of the areas they had difficulty with in the Virtual Case File, and it followed us through into Sentinel. We should have performed business process re-engineering to relax the number of access controls that we had or combine some of these controls, but that just never really happened.

I think in the end, we tried to copy, basically, the rules that were in ACS, and that wasn't successful. Performance suffered. You just can't produce an attribute-base system in commercial software that could provide the kind of fine-grained control we needed. It's a very difficult problem.

FGIT: In your article, you do talk about reducing the number of forms through business process re-engineering.

Israel: Yes, so that was one area where we did business process re-engineering . The FBI has hundreds of forms. More than 800, according to one account. Just as the iPhone has as app for everything, the FBI has a form.

We managed to get the number of forms that Sentinel would be interested in down to around 120, and that seemed like quite an accomplishment—going from many hundreds down 120. But, we only contracted to do about 22. Producing these forms, as I say in the article, was an extremely difficult matter. We had an in-house forms package that we tried to use and for stylistic reasons we didn't like it, we didn't end up using it. We went out and bought a forms package that we thought would work better, and there were some problems with it. So we ended up writing our own forms. Lockheed wrote Java server pages. And that eventually was successful, but how many forms we actually ended up with, I don't think we even got to 22.

The same with workflow. For these forms, there should be a complementary electronic workflow. The workflow allowed you to route the form through an approval process before it eventually went to the database.

FGIT: And it's here in the article where you talk about some of the difficulties the FBI had in implementing the service oriented architecture.

Israel: SOA was the buzzword of the day. It was really big back then; you could set up a number of independent services and define interfaces and be able to call on those services with other applications, going through a common bus.

In my opinion, we didn't implement that. When you get right down to it, you fall back on what you know and what tends to work best, and that's basically building a three-tiered environment: a database, middleware and a portal on top. You end up doing a lot of hardwired software integration. Did we implement a pure SOA environment? I don't think so. It was just too elusive.

FGIT: Was it that the SOA standards conflicted?

Israel: I don't think this is necessarily the FBI's fault. I think it was more industry's. The industry was still growing at the time when it came to service oriented architecture. We bought a number of commercial applications and hoped that all the functionality of the application was exposed through an interface. But what you find out in reality is that only a limited amount of functionality is exposed. It was much more difficult to build a pure service oriented architecture with web services than anybody ever imagined.

FGIT: The difficulty with SOA and software integration, in general, was one of three main problems – three wicked problems – that you said you encountered?

Israel: Yes. The question is, how do you integrate all these commercial applications? If you listen to what everybody was talking about at that time, conventional wisdom was that you set up an enterprise service bus and basically tie services together through web services. It was almost as if you could plug and play applications into the bus. The reality is that that kind of integration is extremely difficult. Just trying to integrate one company's workflow application with another company's document and records management application either with or without a bus was a nightmare. Eventually, we had to give up and use the workflow built into the document and records management application.

FGIT: But the ESB did produce an early win in Phase 1?

Israel: Yes, they successfully used the ESB early on to tie into the mainframe. So, in that sense, it acted as an interface to the mainframe. The first phase of Sentinel was successful in the sense that it provided a web-based interface to some of the functionality in ACS.

But the next phase was to build basically an independent electronic case management system, and that's where this whole thing started unraveling.

FGIT: Reading your article, it looks like people were convinced that Phase 1 was a success, so they looked forward to a success in Phase 2, and not only that, doing it faster.

Israel: Yes, but there were several issues in Phase 2 that led to our demise. One was the technical problems in building an independent electronic case management system. The engineering issues—we already talked about the security aspects, with the fine-grained access control. There were also migration issues – migrating data out of the old system into the new system. And then, ACS has 40-plus databases attached to it. Migrating off those databases into the new environment was another aspect of the migration problem.

But there was another problem, too, which I spend a lot of time discussing in the article. And that is that a project of this kind, to be successful – if you can be successful on these large projects – to have a chance, you've got to have some pretty savvy engineering in the agency.

What I try to argue is that I don't think federal agencies can build systems any more sophisticated than their engineering base. Our IT engineering base at the FBI was limited, very limited. What we tried to do, and it was abetted by the Office of Management and Budget, is we tried to make up for this lack of engineering talent with very strict program management.

And it's still that way today – agencies must have strong program management! Agencies must have people who are federal acquisition-certified, or, in DoD, DAWIA- certified. And what happened in the bureau is that we got all these people certified. They went to program management courses, and they got their certifications – some of them worked up to the highest levels. But very few of these people had any engineering experience.

Very few of these people had ever written a program, written any code. Very few of them have, let's say, established, set up, or created a database. Very few of them knew configuration management, or how to perform software testing. So you've got this major software engineering project going on, and in the FBI, we just didn't have the expertise to pull it off. So what we did was overcompensate with program management.

We program managed IT programs to death with Earned Value Management and costs and schedules that PMs put in place. But without proper engineering understanding, all of these efforts fall short.

FGIT: You also say in the article that the Office of Personnel Management classification for IT workers [2210 Information Technology Specialists] may be part of the problem.

Israel: One of my arguments is that we've got to have stronger engineering in government. But people will look around and say, well, according to OPM, in 2010 we had 77,000 information technology specialists in the government. And that sounds pretty good. But when you dig down and look into what types of people are in the IT field, it ranges from people with no degrees at all – for example, in the FBI, we had secretaries who became operational computer specialists, people who maintained a computer at a very basic level or worked at a help desk –to those who have Ph.Ds. in computer science or computer engineering.

The ITS field has so many kinds of backgrounds and degrees that if Congress wanted to know if an agency had the basic engineering skill to pull off a $450 million software engineering project, there's no easy way to tell because people with very different computer backgrounds are lumped together.

It turns out that in the FBI, the number of really good software engineers is really limited, like it is in most federal agencies. That 2210 classification makes it hard to determine just how technical, how sophisticated engineering-wise any given agency is.

What I recommend is that we have to spend more time classifying people into the proper career fields. There is a computer engineer career field, 0854, and there is a computer science career field, 1550. This is a start. Then, you'd begin to know, does agency X have the personnel who really have the academic credentials and the experience to take on a large software engineering project.

FGIT: Just to ask my devil's advocate question, you have program managers who say, "I don't *need* to know about the technical stuff, I know how to program manage, and engineers don't."

Israel: I don't think that's quite the case. Let's take the program managers, for example. You get into a project that's complex, like Sentinel, and maybe you're a little computer savvy, but not terribly, and now, all of a sudden, you've got problems that relate to the integration of software. You've got software testing priority and severity issues, software bugs popping out of the woodwork. You've got business process re-engineering issues and on and on. I would argue that unless you have engineering insight as a program manager, you're not going to have the faintest idea what's going wrong with your program. All you know is that cost and schedule are off, but you don't really know why. And it's digging down into that underlying *why* that is of critical importance.

I would argue that an engineer who is trained in program management is a more effective leader on a software engineering program than a program manager who's trained in language, political science, or business administration.

FGIT: In your article, you also say the director called bimonthly Sentinel meetings. And there was a graphical display of the program status. You say that from meeting to meeting, the temperature on the thermometer that expressed the overall status [in red, yellow, or green] was always yellow, trending toward green.

Israel: Right. It was very frustrating, especially when we were several years into the program. The program started around 2005. Let's say now we're getting into 2008 or so. The second phase, which was supposed to deal with building an independent case management system, some of the groundwork that was needed to do the second phase, such as building a security model, understanding how we were going to migrate data, how we were going to integrate commercial software – that kind of technical understanding was generally missing.

I mean, some of us anticipated the wicked problems, but what happened is that a lot of the really hard things in Sentinel were bow-waved to the end of the program.

Like data migration – there were at least four different increments in Sentinel devoted to data migration. It kept getting put off and put off. And yet, the program managers, they would come to meetings with the director, and it would look like everything was going swimmingly.

I'll just put it out there like it is – that frustrated the hell out of me. I just couldn't stand it. I stopped going to some of the meetings because I just couldn't believe that we had these major issues going on, technical issues, and yet as far as the program managers were concerned, we were yellow, trending toward green.

FGIT: Isn't EVM supposed to produce reasonably accurate, reasonable recent depictions of program status?

Israel: I'm not a big adherent to earned value management. I don't put a lot of faith in it. And the reason is that government is notoriously bad at predicting how much these projects are going to cost, and how much time they're going to take. You have to have incredible engineering insight and experience in order to come up with a realistic cost and schedule for complex projects.

I make the argument that in government, when we spend so little time prototyping, building prototypes for the very hardest problems, for risk reduction, that you miss an opportunity to improve earned value management, because you would know through prototyping just how hard something is going to be, and how much time it's going to take.

In general, I think earned value management is really pretty poor. It's not a good indicator, because we do such a horrendous job of estimating cost and schedule.

The second thing is that it was actually determined in the FBI, that during phase 1, the earned value management was manipulated. The costs were going up, and so the PMO would change the schedule goal post. The reason someone caught on to this is because when we came in for the biweekly meetings with the director, the earned value management was *perfect*. It was following the projected line perfectly.

Statistically speaking, that just doesn't happen. One of our assistant directors, who is very savvy on statistics, got all the cost information and determined that Sentinel was dorking with the goal posts.

This revelation also came out in the inspector general's reports that the DOJ has been putting out since the beginning of Sentinel. This is not new.

But after that, as we moved into the next phase of Sentinel, the earned value management was looking good. But underneath the hood, we weren't doing the right technical things to advance the program.

Instead, we spent time in second phase screwing around with things like building configuration management and automated patch management capabilities – all of these things were fairly straightforward. They were important, but they didn't advance case management at all.

The program manager could say, 'Hey, we've have had success, we're doing great here,' but we weren't doing the right things. We were bow-waving the hard stuff.

So it's not until you get to the end of the program that things start to blow up. I think that's why earned value management was not a good indicator. It didn't detect or predict this.

FGIT: Does EVM drive the bow-waving, or are there other factors, as well?

Israel: No, EVM didn't drive the bow-waving. It seemed like we made things increasingly easier for Lockheed Martin in the second part of the program. We allowed Lockheed to focus strictly on administrative case data – memos and the day-to-day business of the bureau – while work on terrorism cases, criminal cases and others were put off till later.

We allowed Lockheed to work on admin cases, which were easier, not nearly as many access controls, in the hope that they would succeed, and we put off the harder stuff.

FGIT: Getting back to what happened after phase 1's success. It looked like Sentinel's succeeding. Was it then that it was decided it should be sped up?

Israel: We really didn't speed it up, but after phase 1, which was basically an 18-month project, senior management got very concerned that the next phase was going to be really hard and long. What if Lockheed didn't make it? Do we have to wait another 18 months to find out that they totally screwed up? The thinking was "Let's break this project up into smaller increments and have them deliver these increments every 6 months or so." Then we would have a better read on whether they were going to be successful or not.

FGIT: Is that sort of what you advised in the beginning of the interview – smaller increments?

Israel: I see what you're saying. I think that if we had started from the beginning with a plan to break things up into smaller chunks, then yes, that's probably the direction we should have gone. The problem with Sentinel is that the chunks they broke phase 2 up into did not tackle the hard problems immediately. The PMO put them off into the distance. One of the first chunks was, like I mentioned earlier, patch management. Another chunk was configuration management. Another chunk was system administration tools, another, migration – but instead of working on migration, they put it off until later. This kept happening.

FGIT: Is "chunking" synonymous with "agile"?

Israel: No, not really. When I speak of "chunks" I'm thinking more of a spiral, incremental type of development. Agile development is something the FBI moved to in 2010. Basically they were trying to perform agile development in Sentinel with 7 - 30 day sprints. You produce a little software, you test it, you go to the customer and say, "Hey, how is this?" and if things are good, you go on to the next step.

FGIT: The bureau has emphasized a lot that by adopting agile, that's the way they're going to get the program under control. Is that a way of indeed getting it under control?

Israel: I've been gone since October 2009. I've kept up through contacts and basically the inspector general's reports. The program has been going on for 8 years now, and there's also been a lot of press coming out from the former CIO and the FBI that "The project is under cost; we're on track to deliver it this summer," and maybe they will. Maybe they'll pull off a miracle, but I think that just because you switched to agile, or just because you did this or that, you're not going to be able to deliver this program and meet the requirements that were there at the beginning.

My opinion with IT is that if you do not lay the foundation correctly, everything that you build up above – all the different floors and rooms – will have a lot of cracks.

FGIT: One thing you do suggest is using a CMMI model for government agencies.

Israel: What I'm trying to suggest is that if I were Congress, I would look at agencies in disbelief and say "You want $400 million? Prove to me that you can handle that kind of money on an IT project." I would look at the number of software engineers, database engineers, how many of these people does the agency actually have? I'm not saying that agencies should have hundreds of them, but you have to have *some,* especially at the program management helm, to know what's going on inside your projects.

How many people trained like that do we have in government? I think the answer is very few. But agencies haven't focused their hiring with this in mind. They need to. The second thing is that once you know how many engineers you have, Congress needs to look at "What have you taken on in the past? What kinds of capabilities have you successfully built? What kind of maturity do you have in developing software projects, network projects, different kinds of engineering projects?" That's where you look at a maturity model. It doesn't necessarily have to be CMMI, but a maturity model that says "This agency has done pretty well with the money they have received in the past, and they have pretty good engineering savvy, a pretty good experience base, and yes, I think that they can probably handle $40 million a year for an IT project."

FGIT: Is there any agency that's capable of that kind of self examination?

Israel: Very few. I think with the help of OPM, you must start to reclassify your computer personnel, the people who work as ITSes. And you might even look at creating more categories than just 0854 and 1550.

FGIT: You also talk about prototyping – there's been some talk about trying to increase it in government. Is it fair to say that, at least up until you left in 2009, you didn't see enough, or any of it?

Israel: I saw very little. Prototyping was something that we should have done at the beginning of Sentinel. There were recommendations from committees that commented on Sentinel and on our IT progress that we should build prototypes. It didn't happen. Too much time and money. We just jumped into the project and let the contractor figure it out.

More prototyping should be done in the future, but one of the key issues about prototyping is that you need space, you need an environment. That kind of thing was always very difficult at the FBI. We're in a building that's nearly 40 years old. And trying to find space to build a lab environment is very difficult. But you could consider doing something like this on a larger governmental basis – building prototyping labs in the Intelligence Community or within law enforcement. And with virtual machine technology it's much easier to walk in (or login), build your prototype, and when you're done, erase it and go home.

For more:
- download Israel's article on Sentinel development from IEEE's *Computing Now* (.pdf)

Related Articles:
FBI delays Sentinel rollout to May 2012
Report questions FBI's ability to implement agile development for Sentinel
FBI: We'll complete Sentinel with $20 million and 67 percent fewer workers

Source URL: http://www.fiercegovernmentit.com/story/qa-jack-israel-fbi-sentinel-and-federal-it-development-shortcomings/2012-07-01