FBI Sentinel Is In Trouble

Chuck Connell, BeautifulSoftware.com

The U.S. Federal Bureau of Investigation has been trying for twelve years to modernize its case file computer system. The current system was antiquated when it was introduced in 1995 and has been widely criticized as one of the reasons the FBI was not able to "connect the dots" in its terrorism investigations in 2000 and 2001.

The first modernization effort, Virtual Case File, ran from 2000 to 2005, cost $170 million and failed completely. This project was a classic tale of all that can go wrong with software development. I used it as a negative example in a recent article titled *Why Bad Things Happen to Big Software Projects* and in my book, *Beautiful Software*. Harry Goldstein wrote an excellent, detailed post-mortem, *Who Killed the Virtual Case File?*

In March 2006, the FBI started the Sentinel project to try again. The original target completion date was December 2009. The target slipped to June 2010, then September 2010, then September 2011 and then January 2012. Recently, the completion target slipped again to May 2012. The cost has also risen, from $425 million to $451 million, dwarfing VCF. And it appears this latter figure may be disingenuous, as some required costs are not included in the total.

Sentinel will not succeed, given its current definition, budget and schedule. I define "success" as working with all its intended features, for all users, with high quality. Even with a more forgiving definition of success – allowing for minor missing features, decent quality, an additional 10% budget increase, and a four-month slip to September 2012 – Sentinel will fail. I strongly suspect that in May 2013 it still will not be fully operational with high quality.

How do I know this? I have studied, taught, practiced and written about software project management for 30 years.[1] I have read the last four Sentinel project audit reports from the Department of Justice's Office of the Inspector General (OIG), which were critical of Sentinel. Each audit includes responses from the FBI CIO, who is overseeing the software development, presenting the FBI's take on criticisms from the OIG.

[1] Full disclosure: I am a software project management consultant. I called the FBI CIO and left a voice message offering my help, for a fee, to devise a Plan B when Sentinel does not work in May.

Overall, the OIG audits paint a picture that is all too familiar for failed (and failing) software projects.

- Many schedule slips, each one promised to be the last.
- Increasing budgets, each one final, with some possible slight-of-hand to hide the true numbers.
- A development team so wrapped up in their project that they do not see the forest ("this project is off the rails") for the trees ("we just need to tweak the delivery date for Phase 2 Segment 5").
- An attempt to deliver a finished solution "all at once" to users, because there is no time to gradually introduce larger and larger portions of the software to the user community.

Anyone who has studied project management has seen this kind of troubled software effort before. It has happened many times to many organizations, including the FBI. The OIG audits have the unmistakable feel of "here we go again". In addition to the concerns listed in the audits, there are other problems with Sentinel that struck me as I read the reports. These issues are somewhat hidden between the lines of the audits but deserve attention.

Goal of Agile

In September 2010, the FBI fired the prime contractor for Sentinel, Lockheed Martin, for poor performance and moved development in-house. At the same time, they switched to the Agile programming model to increase development speed. But the main goal of Agile is not speed; it is to more closely align the finished product with what users actually desire.

Agile was created as a response to the common software problem of users proclaiming, "Now that I see the software working, I realize that I got exactly what I asked for, but it is not what I want!" Agile attacks this problem by recognizing that all of the requirements for a software project cannot be known at the start, so the method embraces change as the project moves along. Agile saves time to the extent that it eliminates the need to re-write large parts of a system upon discovering users don't like them.

But there is no evidence I am aware of, or logical reason to suppose, that Agile programming is a faster way to deliver a known, fixed set of features. If you already know what you want, Agile is probably slower because there is a lot of overhead for running the project in that style. The FBI is to be commended for adopting a modern software development model, but should not fool itself into thinking that Agile is a magic bullet for speed.

Form-Based Architecture

The software architecture of Sentinel is based around 18 forms that the FBI uses to do its work. These forms were originally on paper and are now a combination of paper and outdated computer files. Titles of the forms include "Report of Information That May Become the Subject

of Testimony", "Intelligence Bulletin" and "Search Results Document." One of the key goals of Sentinel is to transition the forms to a single electronic format. The software development schedule – deliverable items from the programming team to users – is based on the forms. Some forms were delivered during Phase 1 of development and more during Phase 2. Other forms are forthcoming in Phases 3 and 4, etc. A potential problem for Sentinel is that the number and contents of the forms will likely change over time.

For Sentinel to be flexible enough to adapt to future investigations, its architecture should not be rigidly tied to the current 18 forms. It is important that Sentinel be able to accommodate modest changes to an existing form, with incremental effort rather than a significant rewrite, and be able to create a new form with incremental work. I have not examined the Sentinel programming code, but based on comments about the code from both sides, it appears the software may be too coupled to the existing forms as they are today. If so, Sentinel runs the risk of not being sufficiently flexible to adapt to reasonably foreseeable future needs, even if it is some day finished.

(The standard solution for this problem is meta-form architecture, such as XML, that allows naturally for changes and additions.)

Hardware Infrastructure

In October 2011, the FBI ran a test of the partially completed Sentinel software, with several hundred users testing it for a four hour period. The users appeared to like some of the software interface and features, but the test was a failure due to various hardware/network response times. The FBI is currently replacing that hardware to correct the problem.

But what are they replacing? The audit, and the FBI response to it, does not say. Is it servers, workstations, the network backbone, a storage system such as a SAN, network cards in workstations, or something else? Most of these items are a big deal and require time to research, purchase, install and configure. Whatever is being updated must be done by April 2012, when the test will be repeated. If there are any problems during this test, there will be no time to fix them, since the final deadline is just a month later.

The late stage of development of an important software project is not the time to be changing a large organization's computing infrastructure.

Agile in Use?

There is something else disturbing about the October test however, beyond finding that current hardware is inadequate. The fact that this test took place at all means the FBI is not actually using Agile development. One the key tenets of the Agile method is "working software". This means that frequently, every two weeks or so, the development team releases fully working software (albeit with missing features initially). Instead of waiting years to see a functioning

product, there is always a recent operational version of the software available. But "working" means that it really, really works and (with incomplete features) is ready to be installed at customer sites.

If the FBI discovered in October that their software builds would not run adequately on the standard agency computer/network infrastructure, then they have not been releasing working software. Most likely, the frequent software builds they were producing ran in a limited test environment. This is not Agile development.

90% Complete?

The FBI claims that Sentinel was 88% complete on December 6 and that more has been accomplished since then. But this cannot be true. For a software project to be 90% complete, all of its major features must be working well, it must be rolled out to a substantial number of users who have thoroughly exercised its operations, the bug list must be largely burned down and the development team must be performing final "fit and finish" changes. None of this is true for Sentinel.

More substantially, saying that a software project is X% complete is well known to be an unreliable metric. All it means is that 90% of the items written on a development checklist are now checked off. The remaining items may very well be the hardest. Or, as often happens, the checklist does not even include important work items which have not yet been discovered.

**

The sum of these concerns, those expressed directly in the OIG audit reports and those I surfaced above, lead me to conclude that Sentinel is a deeply troubled software project. This time, half a billion dollars is at stake, as well as another potential setback in the FBI's ability to do its job efficiently.

Chuck Connell knows software projects from all sides. He has been a hands-on programmer, a software product architect, a manager of programmers, a university teacher of software engineering and a consultant. His consulting practice BeautifulSoftware.com helps organizations assess the health of software projects, manage ongoing projects and turn around troubled projects.

For more information….

http://www.chc-3.com/pub/bad_things_big_software.pdf *(Why Bad Things Happen to Big Software Projects)*

http://www.amazon.com/dp/1456438786/ (*Beautiful Software*)

http://spectrum.ieee.org/computing/software/who-killed-the-virtual-case-file (*Who Killed the Virtual Case File?*)

http://www.justice.gov/oig/reports/2011/a1208.pdf (*December 2011 OIG audit of Sentinel*)

http://www.justice.gov/oig/reports/FBI/a1101.pdf (*October 2010 OIG audit of Sentinel*)

http://www.justice.gov/oig/reports/FBI/a1022.pdf (*March 2010 OIG audit of Sentinel*)

http://www.justice.gov/oig/reports/FBI/a1003_redacted.pdf (*November 2009 OIG audit of Sentinel*)