Name: Luke Chung
Title: President and Founder
Organization: FMS, Inc. (www.fmsinc.com) and FMS Advanced Systems Group (www.fmsasg.com)

Testimony

Summary

About Me and FMS, Inc.

I'm the president and founder of FMS, Inc., a privately held software development firm in Vienna, Virginia. For 27 years, we've created database solutions with a combination of commercial products and services. In response to 9/11, our Advanced Systems Group created Sentinel Visualizer, a product for the counter-terrorism, defense, and law enforcement communities that led to our only outside investor, InQTel, the CIA's venture capital arm. We have tens of thousands of customers in over 100 countries, including 90 of the Fortune 100. Our Professional Solutions Group has created a wide range of custom solutions, some which are more complex than Healthcare.gov, but never more expensive. I'm a graduate of Harvard College with a bachelor's in engineering and a master's in physical oceanography.

My Experience with Healthcare.gov

On October 1, I visited Healthcare.gov to get an insurance quote for my family. The experience was so terrible that I documented the technical problems I encountered and wrote a blog post about it. I could tell immediately from the nature of the crashes I encountered that the site was not ready for prime time. It had a terrible design that was not consumer friendly, seemed to be coded by amateurs, and wasn't tested. I could tell the site would not work for one person much less the expected load.

The blog post I wrote on October 1 went viral as people began to understand the problems were deeper than too many users. That led to being quoted in the New York Times and appearing on radio and news shows such as CBS, CNN, Fox, MSNBC, NBC, Hannity, Greta, Al Jazeera, Geraldo, etc. Throughout the period, I've learned more about the website and its many problems both political and technical.

Healthcare.gov Overview

This web site should not be that difficult to build. It doesn't provide healthcare. It doesn't even provide health insurance. It's comparing plans and applying for a subsidy. It's the automation of a paper form.

Security Implications

Security is considered at the beginning of a project, not at the end. Avoiding the collection of unnecessary personal information is the first step to reducing security issues. Separating the user experience from backend legacy systems is another. The pressure to make a software solution "work" is not conducive to good security. There are ways to improve the user experience, scalability, and security.

Contractor Abuse of Taxpayers

Healthcare.gov is just one example of a software project gone awry that government contractors profited at the expense of taxpayers. I originally thought the web site was created by people who didn't know what they were doing; that they were trying to do too much in an unnecessarily complicated and thorough manner. My thoughts have evolved and I now feel that it's designed quite cleverly to maximize taxpayer expense. This is a scandal that needs to be investigated. Follow the money and I believe you'll see design decisions that led to increased costs. There are ways to improve governance to fix this.

Background

Thank you for inviting me to your hearing.

About FMS, Inc.

I'm Luke Chung; the president and founder of FMS, Inc., a privately held software development firm located in Vienna, Virginia. Since 1986, FMS has provided software products and development services to commercial and government agencies. Over 27 years, we've created a wide range of database solutions helping organizations make better decisions based on data. These important decisions include delivering services, managing operations, understanding finances, increasing accuracy, improving customer service, making fewer errors, targeting criminals, making more money, and increasing efficiency. We have tens of thousands of customers in over 100 countries.

In the 1990s, we became the world's leading provider of commercial products for Microsoft Access with 12 solutions to help people better analyze data, automate email blasts, create better solutions, eliminate errors, and provide system administration.

In response to 9/11, we created the FMS Advanced Systems Group to use link analysis and social network analysis (SNA) to find hidden relationships among people, places, and events. That led to the creation of our Sentinel Visualizer product that helps analysts in the counter-terrorism, defense and law enforcement communities, both in the US and abroad. Sentinel Visualizer led to our only outside investor, InQTel, the CIA's venture capital arm.

In addition to our commercial off-the-shelf products, the FMS Professional Solutions Group has created custom database applications for a wide range of customers. Examples include the Logistics Support System for the Pan American Health Organization sponsored by six UN Agencies. It coordinates humanitarian relief logistics for disaster zones and is deployed with language localization features in over 100 countries, including the Philippines. FMS also created a course management system for the Defense Acquisition University, which provides non-military training to all branches of the DoD. FMS has also created custom solutions for event management, e-commerce, logistics, education, healthcare, public works, nonprofits, and businesses.

About Me

I'm originally from New York, grew up in Orlando and Sarasota Florida, and am a graduate of Harvard College. I have a bachelor's degree in engineering, and a master's degree in Physical Oceanography. Prior to founding FMS, I worked as a management consultant at Strategic Planning Associates/Mercer.

- Current member and past president of the Washington DC Chapter of the Entrepreneurs Organization.
- Serve on the Business and Community Advisory Council to the Fairfax County Virginia Public School Superintendent.
- Serve on the Information Technology Policy Advisory Committee to assist the Fairfax County Board of Supervisors oversee county technology investments. The committee exists because the

supervisors recognized years ago they were unable to provide the proper governance over their technology investments.

Caveats

My testimony is based on my personal experiences and opinions. I am an observer to the Healthcare.gov web site and am not personally involved with its design and development. Any suggestions of incompetence or wrongdoing are comments intended for further investigation by the committee.

My Perspective

I am providing my testimony from a non-partisan perspective focused on my decades of experience creating database solutions, the challenges of running a small business, and having observed how the government contracting world works.

In 27 years running FMS, I've experienced multiple government administrations, economic cycles, and changes with technology. I run a small business and have responsibilities to my clients, firm, employees, and family. These obligations include buying health insurance.

Experience with Healthcare.gov on October 1

On October 1, I visited the Healthcare.gov web site to get an insurance quote for my family. I wanted to see what policies were available and how they compared in features and price to what my small business is currently purchasing in our group plan.

What started as a simple shopping experience turned into a venture inside the technically worst web site I've ever visited. It was so bad that I started documenting the bugs I encountered. I was shocked because the mistakes were so amateurish that it seemed the web site was created by people who had never been paid to write commercial software. Based on my experience, I realized that if those types of bugs existed, the web site had huge problems way beyond the number of users. I sensed that it would not support one user, much less the millions expected.

The shocking part is that this web site should be very simple:

- It does not provide healthcare
- It does not even provide health insurance
- It's supposed to let consumers compare and choose among insurance plans
- It's supposed to generate a subsidy, if any, to buy insurance
- It is essentially the automation of a 12 page paper form

I shared my findings in a company blog post entitled **Healthcare.gov is a Technological Disaster** (http://blog.fmsinc.com/healthcare-gov-is-a-technological-disaster/) – See Appendix A. It includes screenshots of the crashes and suggested that I was embarrassed for my profession for delivering such junk. It looked like the developers never used or tested it. I concluded that the quality of the work

wouldn't pass a computer science class and that there would be huge Public Relations problems that could doom the entire Affordable Care Act. That's what I saw on Day 1.

Response to My Blog Post

While the contractors and administration tried to spin the problems as the result of too many users, my blog post – which provided a non-partisan, technical evaluation of Healthcare.gov – started getting picked up by multiple web sites. And through the power of social media, it went viral.

Within a week, I was quoted in a New York Times article which was followed by interviews with radio and national TV news channels including CBS, CNN, Fox, MSNBC, NBC, Sean Hannity, Al Jazeera, Greta van Susteren, Geraldo Rivera, etc. It has led to this testimony.

Offering Solutions

Since I don't like being a critic without offering possible solutions, on October 14th, I wrote another blog post outlining how Healthcare.gov can be properly built: **Creating a Healthcare.gov Web Site that Works** (http://blog.fmsinc.com/creating-a-healthcare-gov-web-site-that-works/) see Appendix B.

My suggestions would a web site that would better address the needs of the customer, be simpler to develop, easier to test, more robust, support more simultaneous users, and be more secure. It would separate the shopping experience and an estimate of a subsidy from the actual application to receive a subsidy (the part that needs to be secure). The marketplace would be the central site where it would be easy to compare insurance plans before worrying about pricing and subsidies. The site would be hosted on commercial cloud providers that could scale to support huge numbers of simultaneous users. It would use commercial business software that would significantly reduce the amount of code that needs to be written and tested, which would also reduce the security risk.

Healthcare.gov Observations

Here are my observations about the technical issues I encountered on the Healthcare.gov web site:

- It's poorly designed. It doesn't address the needs of a consumer trying to shop for something, nor is it designed to support lots of users or high security.
- It's poorly developed. The site has such amateurish errors that it appears to be created by inexperienced developers.
- It's not tested, or if it was tested, the test plan was woefully inadequate.
- In my experience, encountering that many bugs in such a short period of time indicates that was only the tip of the iceberg with many more bugs below the surface. As bugs are fixed, more bugs will be found since those sections were never adequately tested before.
- The management team and contractors seemed to think the site was production quality on October 1. It clearly wasn't, which would indicate that those people don't understand what production quality means. They shouldn't be involved with the project since we've experienced what they consider shipping quality. I do not consider what was delivered to be beta (test) quality.

Security Implications

Lack of competent technical oversight not only leads to waste, but to potentially devastating security vulnerabilities if complex systems that millions of people depend on are undermined or brought to their knees by attackers. Technology alone cannot deliver security, and the more complex a system is, the harder it is to secure against known threats, much less unknown ones which are sure to emerge in the future. When developers operate under deadline pressure, they tend to cut corners to "just get it to work", generating fresh security vulnerabilities and bugs.

- Nothing is ever perfectly secure.
- Security has to be considered at the beginning of the project, not at the end.
- The most important part of security is to NOT collect secure information unnecessarily.
- The next step is to minimize the places where security is necessary. The sections in which users shop for insurance policies, get an estimate of the subsidy, and buy a policy without a subsidy should not require any security.
- Another design consideration is to create as few places of vulnerability as possible. That means fewer screens, fewer places where data changes hands, and running secure processes offline separate from the user interface.
- The skills to build a secure web database application are far more advanced than the skills the existing developers failed to exhibit. A chain is only as strong as its weakest link.

Contractor Incentives

Originally, I thought the design decisions of the Healthcare.gov site were done by amateurs who didn't know what they were doing. I'm now moving away from that conclusion.

Instead, I'm seeing how the design decisions may have been made to maximize taxpayer expense and vendor profitability.

Government Contractors

The current government contracting system excludes technically-qualified companies by making it difficult for them to bid and work on government projects. The companies that specialize in government contracts are good at winning government contracts, protesting lost awards, and creating change orders. They are not known for their technical expertise. Their strategies and operations would not be competitive in the private sector.

Currently there is no downside for failure to deliver on a government contract. There is nothing to prevent failed vendors from bidding on future projects or being suspended from existing projects.

Abusing Taxpayers

I don't know how the decisions were made, but if I look at it from the contractors' perspective with the knowledge that the budget was essentially unlimited, it would explain how choices were made to add complexity, increase billable hours, purchase more hardware and bandwidth, and maximize profits.

Of course, the big mistake was not delivering a quality solution. Unlike many other IT projects that have failed in the federal government, this one let the public experience the quality of the deliverables.

Examples of areas that maximize profits:

- Performing an identity check for each visitor. Is the credit agency paid for each check?
- Creating a user login in three screens rather than one? Was the contractor paid per screen? Was there consideration that more screens use more resources? Why ask for secret questions?
- The email confirmation process requires almost immediate confirmation. My 30 minute delay in responding canceled my account and required creating a new login. Why does this feature exist?
- Why are the screens to fill out the application one question per screen? Why not put all the questions on one screen to minimize the complexity, data exchanges, and improve scalability and security? Were contractors paid based on the number of questions and screens?
- Why ask optional questions such as race that are not part of the subsidy process?

Addressing Contractor Complaints

From what I can see, the contractors are trying their best to deflect blame:

- There are claims the government was changing the design at the last minute and there wasn't enough time for testing. On every project I've worked on, designs are always changing and there has never been too much time for testing. It's the responsibility of the contractor to provide the guidance and services to ensure success.
- There are claims that individual portions were working but the overall system was not. Based on what I observed, the web site wasn't working even if the overall system wasn't tested. My belief is that both the individual portions AND the integrated system were not working.

Where did the Money Go?

I don't understand how the contractors could have charged the taxpayers so much money. At $200 million at a generous $200 per hour, that's 1,000,000 man hours. That's 500 man years. Now the numbers are even larger. Where did all that time go?

Technology Management Recommendations

This is a complete breakdown in managing technology investments. People do not understand when a project should cost $1 million vs. $100 million. In the private sector, a $1 million budget to build a web site is huge. The government needs to remember that buying from companies that specialize in government contracting is not the same as vendors who are competitive in the private sector.

Create a Technology Accountability Office

A Technology Accountability Office (TAO), a non-partisan entity similar to the GAO that is capable of assessing and managing government technology projects. Policy makers, politicians, and bureaucrats do not possess the technology skills to keep up with the rapidly changing technology options. They also don't understand what technology should cost or the implications their decisions have on cost, security, and other options. My serving on the Fairfax County Technology Policy Advisory Committee is an example of this type of governance.

Enforce Accountability

Past performance is considered an important part of winning government contracts but it doesn't seem to prevent contractors involved with failed projects to continue winning new contracts. If qualifications matter for selecting contractors, when do contractors ever get permanently banned? Multiyear or permanent bans should target underperforming vendors to prevent them from bidding on new contracts and removed from existing ones.

In the private sector, vendors that fail would rarely be allowed back. Do we have a too big to exclude policy?

Audit and Investigation Needed

An exhaustive investigation and audit of the Healthcare.gov project would help determine the various points of systemic failure in order to ensure that a debacle of this magnitude never happens again.

Experience of the Development Team

The experience of the vendor is important, but what's most important is the experience of the people actually doing the work. Given my sense that the developers were quite junior, it would be interesting to learn their previous experience building commercial database web sites, what they were being paid, and what the taxpayers were charged. Make sure people involved with the entire life of the project are questioned, and not just the ones remaining today.

Development Management and Environment

- How were the deliverables designed, scheduled and delivered?
- How were the teams managed?
- What code reviews were held, and by whom?
- What development, testing, and staging environments were employed?
- Was there a test plan? If so, what were the results of the test plan before October 1? What bugs were considered acceptable for deployment?
- How did the test plan change and who was paid for the October 1 that was so bad?
- Is load testing and balancing in place?
- What kind of security reviews, threat analyses, and mitigation strategies were undertaken?
- What kinds of security vulnerabilities were detected, and when are they scheduled to be addressed? How are security issues addressed on an ongoing basis?

Technology Selections

- Why did they take such a strong stand on using open source "free" software rather than commercial business software that would require less customization (and therefore cost less with fewer security vulnerabilities)? (TheAtlantic.com, June 28, 2013, *Healthcare.gov: Code Developed by the People and for the People, Released Back to the People*)
- Why did they create their own cloud rather than using better and cheaper commercial cloud providers? Especially when large portions of this site do not need any security.

Design Flaws and Bugs

Secretary Sebelius and HHS have announced that they've fixed hundreds of bugs, which indicates that there are likely hundreds more yet to be found. No matter how many bugs are fixed, the unintended consequence is that more will inevitably crop up elsewhere in the code base. Is the current web site being redesigned to make it work properly for consumers, or are they instead trying to make the existing flawed design functional? Poorly designed systems are nearly impossible to rescue, and inevitably lead to further support costs down the road. When a complex system is created by multiple vendors with no technical managerial oversight, it is inevitable that systemic flaws will lead only to finger-pointing and recrimination, not to solid, functioning software.

Number of Concurrent Users

The heaviest demand day was not October 1, but will be the day of the deadline to sign up. It's the equivalent of April 15th for the IRS. How are they preparing for that? How many simultaneous users can they support, and what happens if the number of users exceeds that? Is load balancing in place? Are we buying lots of equipment for that one day that will sit idle afterwards? Totally unnecessary if a commercial cloud provider is used.

There are policy implications if the system crashes and people are shut out before the deadline

What Are They Thinking?

- How could they have possibly thought the site was ready to go on October 1? There was a seminar scheduled on HowTo.gov to showcase how the contractors created this great web site but it was postponed due to the government shutdown and later canceled.
- Are they redesigning the web site to make it work properly for consumers or are they trying to make the existing bad design work?

A More Open Policy

- Many companies could have created the Healthcare.gov web site or similar database web sites. Why is it so difficult for technically qualified companies to bid and work on government projects?
- Why isn't the data on the insurance policies, pricing and formulas for subsidies opened in a manner that the private sector can create their own web site marketplaces?

Conclusions

Overall, I'm embarrassed as an American to watch my President and Cabinet Secretary talk about web site design, development, and testing, and promoting 800 numbers. They should be focused on policy and things like Iran and North Korea. Web sites should be taken care of at a much lower level and certainly no higher than the CTO.

The underlying problem of Healthcare.gov lies in the way that government contracts are awarded. Our way of life is becoming more, not less, dependent on technology every day, yet there is no one at the highest levels of government capable of determining when the government is being ripped off.

Taxpayers made a significant investment with the contractors to expect a functional Healthcare.gov web site. While there may be some excuse for complexity with connecting to legacy databases in various agencies, I don't see any reasonable excuse why the user experience would be so defective or the costs so high.

This is a scandal beyond Healthcare.gov and touches on the entire way the government purchases software solutions. Unfortunately, the federal government has paid for even larger software projects that were never functional.

The need for a bi-partisan Technology Accountability Office to investigate and regulate technology at the Federal level is urgent and immediate; not only to stem the hemorrhage of taxpayer dollars, but to ensure the security and viability of the essential systems millions of Americans depend on.

Taxpayers paid Super Bowl ticket prices and were delivered a high school football game. Follow the money.