This Testimony is Embargoed until Wednesday, May 9, 2012 at 2pm

Testimony for House Committee on Ways and Means Subcommittee on Social Security

Hearing: The State of Social Security's Information Technology

William L. Scherlis, Ph.D.
Professor, Carnegie Mellon University
May 9, 2012

Mr. Chairman and Members of the Committee, it is my honor to be here today to discuss the future of one of our most essential American institutions, the Social Security Administration.

My name is William L. Scherlis, and I am a professor and department head in the School of Computer Science at Carnegie Mellon University. My academic research focuses on software engineering and software assurance. I am also acting Chief Technology Officer for the Software Engineering Institute, which is a Department of Defense Federally Funded Research and Development Center (FFRDC) focused on advancing software capability for the DoD and its supply chain. I was at DARPA for more than six years in the late 80s and early 90s. After I returned to the university from that role, I became interested in the challenge of enabling the government to be a smarter consumer of computing technology, primarily focused on Defense but also addressing cybersecurity, e-government, crisis response, and other topics. I have chaired National Academy studies on e-government and, most recently, on software for the Department of Defense. This latter study focused quite closely on the nature of agency leadership for large-scale systems—and why it is important to strengthen that leadership. That is our focus today.

Advancing a new forward-looking vision of service capability at SSA is both a great opportunity and an enormous challenge. However, this is hardly unique to SSA—it is true of most major enterprises. It can be done and is being done in the private sector.

My testimony today focuses on the necessity of architecture-based leadership in information technology (IT) in order to implement a new vision for SSA IT systems, roughly like a large-scale e-business. I consider what that leadership entails and how we might get there.

In the 30 years since many of the existing SSA systems were first stood up, storage capacities, network bandwidth, processing power, and the cost of these things have all improved by between 4 and 6 orders of magnitude. That's a factor of a million. If skyscrapers increased in height by that factor, they would scrape the moon. And we are not just talking about hardware. Software capability and sophistication, much harder to measure, have also improved commensurately. And there is no barrier in sight to continuing this pace—we are nowhere near any kind of plateau.

The ability to manage IT and exploit advances in technology is essential to keeping pace with consumer needs in a constantly evolving, technology driven consumer environment. Recognition of this at the topmost leadership level is essential. The Chief Information Officer becomes a full partner in strategy, and must be empowered and supported through a potentially disruptive period of essential change.

Exploiting the rapid growth in IT capability—hardware, software, and systems—has the potential to dramatically increase productivity and organizational performance at SSA, as with any other organization of scale. At the gross level, macroeconomists attribute 25% of GDP growth and fully 40% of the growth in productivity to Information and Communications Technology (ICT).

Additionally, effective architectural leadership includes the ability to realize major new benefits from a range of emerging technologies. One important example is data analytics. Pioneered by Google and others, including my colleagues at Carnegie Mellon and other universities, data analytics is now being widely implemented in the private and public sectors for business intelligence, health data analysis, national intelligence, transportation management, and many other applications.

I believe it is useful to consider the key drivers of change that can guide a vision for SSA's future, as well as to consider impediments to change that must be overcome in order to achieve that vision.

Drivers of Change

- Obsolescence of systems
- Increased demand for services

In the world of IT, we often hear the phrase "legacy systems." This phrase refers to old systems that operate on obsolete platforms. Most typically, we no longer fully understand them, so these systems are difficult to change and, consequently, are often surrounded by an atmosphere of fear and superstition. Sometimes a few remaining dedicated individuals possess the unique knowledge of the inner workings of these systems. But as managers we may avoid intervention for fear of disrupting some unknown balance of energies with a poorly understood change with unintended consequences. These systems almost inevitably approach a point at which it is no longer feasible, either technically or from a cost standpoint, to continue to maintain them. This is a legacy we would rather not inherit.

An old DoD study indicated that, for long-lived large-scale systems, the costs of reverse engineering—understanding what makes them tick—over their lifespan often exceed the original development cost. This is astonishing, because these are systems the government paid to develop and that it owns. Nonetheless, the failure of managers to "refresh" these systems over time leads to great cost in recovering enough of the lost design knowledge to merely keep these systems going. If a significant event occurs such as a cyber attack or a policy shift, the needed repairs are often impossible, leading to loss of function and a host of potentially unforeseen and undesirable consequences.

Another driver of change is the increased demand for services and opportunities that exist for meeting them.

The most obvious opportunity is modernizing the system that the user sees—web services and their infrastructure. A vision for this is put forth in the document *Re-imagining Social Security* developed by the SSA Future Systems Technology Advisory Panel (June 2010).

But there are also significant opportunities that are less visible to the user, for example taking advantage of modern data-intensive computing. Advances in storage, bandwidth, processing power, software and

algorithms have enabled what we now refer to as "Big Data"—computing techniques that enable rapid analysis and manipulation of vast quantities of data to turn it into actionable information. One could imagine deploying these capabilities at SSA to vastly reduce processing times, errors, and perhaps create a whole new range of user friendly services for the customer and for internal administrators.

A key feature of Big Data is that it involves the distribution of data and processing capability across one or more large data centers. This "distributed architecture"—a way of deploying resources—is enabled by these so-called "Cloud" architectures. Google and Amazon exist because they have mastered and fully exploited this approach. These two firms pioneered these techniques a decade ago, and the techniques are now becoming mainstream both in industry and in a wide range of government applications. But Big Data requires technical sophistication to implement successfully.

What's exciting about the Cloud is that it is fundamentally scalable and incremental. When it is done properly, the technology can be updated in an ongoing manner. If a component in the Cloud fails, software enables us to automatically work around it. On a larger scale, if a large but localized disaster occurs and takes out a number of components, there are ways to work around these as well. This is what DoD calls survivability—the ability of systems to continue to function despite the loss of certain components.

Another feature of this Big Data/Cloud computing approach is the ease of programming. Consider mobile apps—the richness of the software frameworks means that only a small amount of programming is needed to create new capabilities. A similar "framework plus apps" model exists for cloud analytics. Instead of marshaling massive amounts of data to be piped through a mainframe computer, we send small instructions out to where the data is stored. This affords both scalability and flexibility.

The increased demand for and opportunities presented by new technologies are possibilities we cannot ignore.

Impediments to change

- Organizational culture
- Contracting relationships

One impediment to change is a clash between the conservative culture of an organization and the rapid pace of change in the underlying technology. This has as much to do with people as with technical matters. When technology changes rapidly, risk judgments become difficult, and so it becomes easy to fall behind. A series of small reasons not to engage with the new and the innovative—it's too risky!—can freeze a system in the past. Taking risks seems fraught, and it is all too easy just to say no.

The changes specifically related to software and systems architecture are, in many ways, more profound and more challenging for IT leaders than hardware changes. Three examples of profound software architectural changes are (1) frameworks for the cloud and big data, (2) the infrastructure used to create web-based services, and (3) the advent of mobile frameworks and apps. These are all enabled by a combination of innovations in software, in systems, and in architecture.

Technically savvy managers can often anticipate changes in technology infrastructure and respond proactively. Successful firms and IT organizations work continually to stay fresh in their technology as in other aspects of their business. This is a matter of strategy and also of fostering organizational culture. There are many stories of successful firms that paused in this refresh, thinking they had reached a plateau. Many of those firms no longer exist. On the other hand, Intel's Andy Grove was famous for keeping his company successful by causing it to reinvent itself, even in the absence of any immediate perceived threat.

Addressing these shifts can often mean making a series of disruptive changes. Small "smooth" changes to existing systems may seem most conservative, but often transform a patched system into a time bomb of deferred maintenance. The eventual cost of repairs continues to rise, and our aspirations to move into the future become unattainable.

My colleagues have a term for this—they call it Technical Debt. Like unpaid student loans, it won't go away. It just gets worse. In a sense, the various studies of recent years by NRC, FSTAP, and GAO are identifying the nature and extent of the "technical debt crisis" at SSA. Perhaps I'm stretching the metaphor, but we do not particularly want to raise this debt limit. To make matters worse, technical debt can be a self-reinforcing problem. When too much is amassed, it appears even more risky to make the necessary disruptive changes. It is better to have a pattern of well-planned and well-managed small changes that we follow from the outset.

A second impediment relates to government systems and the contracting relationship. Government systems can be difficult to evolve effectively, because the incentives for program managers, contractors, and system managers may not always be aligned with actual mission needs. Even when incentives are aligned, the amassed technical debt can generate fear that precludes effective engagement, especially when we feel it necessary to continue to pay contractors to make incremental upgrades to idiosyncratic legacy systems that only they understand. Indeed, one of the characteristics of technical debt is that systems become increasingly difficult to understand, and in effect have an aura of darkness and fear around them.

Let me say this more plainly. Even the best managers can lose control of their systems. It becomes increasingly difficult to maintain a full intellectual grasp of the systems architecture, the embodied business rules, the key design decisions embodied in the system, and the rationale for those decisions. This is not just a matter of documentation—it is a matter of managing change within the context of a master plan. Successful firms often accomplish this using "agile" techniques, which involve iteration and incremental development. These techniques are employed for a wide range of commercial systems. They are harder to implement at arm's length through contracting relationships, but it is possible and has been done. Teri Takai, CIO of the DoD, has wisely been advancing these practices for use in innovative Defense systems.

The point of these techniques is to accommodate the reality that we don't often have a precise concept of what a system will be called on to do over its entire lifetime. This is never true for commercial systems, since they must respond to constant changes in the competitive environment, the

infrastructure environment, and in strategy. This is also true, quite obviously, in Intelligence and Defense—the threat evolves, the mission changes, the infrastructure evolves, and new technical capabilities must be incorporated.

But it is difficult for some managers to adapt to this model, because it seems harder to manage. Nevertheless, more linear models actually create more risk, not less—the steady growth of technical debt can necessitate responses that are more like emergency bail-outs than the steady stream of small disruptions that keep you moving into the future—just like the small disruptions we all experience when we move to a new home computer or mobile device every few years.

If you look at successful tech-intensive firms—those Fortune 1000 firms that have been around for a number of years—you'll see management patterns that support this model.

Enablers of change

My final topic relates to methods by which we can enable the right kind of change. We are here today as a consequence of a crisis of technical debt. I propose to you six elements of a strategy to address the challenge.

(1) *First, and most essential, ensure there is full commitment from top leadership, and that change is motivated both by need and by opportunity.*

There is danger in passive resistance—those involved with systems must either sign on to the process or move out of the way. I say this because passive resistance can be dangerous in this kind of exercise. Engaging staff to enable them to be comfortable with and conversant in modern technologies, perhaps by visiting organizations that have already made such disruptive changes, can be highly valuable.

(2) *Second, understand what you have—the architecture of the full set of existing capabilities.*

This is the baseline. Existing systems can be assessed by internal and independent experts to understand their architecture and their most essential attributes. This identifies a baseline as well as the most urgent Operation and Maintenance (O&M) issues the agency faces. It must be done in a sufficiently comprehensive way that we learn enough to avoid surprises later in the process. For long-lived legacy systems, this kind of due diligence process can be an enormous challenge, both technical and managerial. On the one hand, shortcomings and difficulties are revealed, but on the other, aspirations are identified.

(3) *Third, develop a cohesive vision of future services, including capabilities and quality requirements related to security, quality of service, and so on.*

This third step can be done concurrently with the second. It includes both developing and articulating a vision of future services. This includes ways to improve the current range of services, such as presented in the FSTAP re-imagining report, but it also includes entirely new concepts such as the Big Data analytics capabilities referenced earlier. It also must consider quality-related requirements, such as security, quality of service, ability of the system to grown over time, and so on.

(4) *Fourth, identify potential changes over the next five to ten years, including scale-up.*

Identify areas for needed variability, where operational and infrastructure features may be more rapidly changing. This can also include identifying potential infrastructure choices associated with established commercial socio-technical ecosystems related to mobile devices, web services, and transaction processing, for example. As part of this consideration, it is essential to understand where scalability is most important. Scalability is what drives enterprises off the mainframe and into the modern data center. Scalability is what lets the enterprise respond to unexpected surges in demand, significant policy changes, and other potentially disruptive activities.

(5) *Fifth, pull these results together and develop a master plan for the overall future architecture along with roadmaps for evolution based on that architecture. This creates the framework of leadership commitment that enables a natural growth process, as is done in other large, IT-intensive enterprises.*

This architecture plan and the associated roadmaps are the principal strategic assets of the IT organization. They manifest the most important commitments that the organization imposes on itself and its contractors. The master plan does not have to be vastly detailed, especially since it will continue to evolve over time, though more slowly than constituent components. The goal is to open the door to opportunities, accommodate them within a natural growth process, and create a framework for consensus among the very top leadership in the organization. This interplay of architecture and strategy is a success pattern in other large enterprises.

A proper architecture development effort contemplates issues ranging from data center architecture to the kinds of analytics and transactional systems that must be incorporated. It considers present and future possibilities for how SSA might engage with end users. It looks forward over a timespan of five to ten years. The conduct of this activity is analogous to a master planning process in city planning. It identifies major directions and constraints, and it builds on inputs from diverse stakeholders. Ideally, it locks in critical quality attributes and enables visionary possibilities.

Architecture is not static, but slowly evolves. Architectural disruptions are most dangerous, so it is essential that the plans be sufficiently long-sighted, anticipating needs and possible opportunities. In other words, like a city, the architecture of a system must allow room for significant and evolutionary growth.

Architecture is the strongest determiner of critical attributes related to security, safety, quality of service, scalability, and the like. We call these quality attributes.

Within this framework, individual projects are undertaken, like the development of apps, which are designed to implement particular capabilities. But overall quality outcomes—the way all the pieces play together—are largely a consequence of the architecture. When this is done well, the increment of effort to manifest new capability or adapt an existing capability can be relatively much smaller than in the more familiar stovepipe scenario.

Managing change within an evolving architectural framework is a hallmark of well managed large enterprise systems. Before that framework is defined, new acquisitions and major projects can impose sunk-cost constraints on the overall enterprise architecture. If we do it right, new efforts can be undertaken with foresight and with engagement with the master planning process.

I do need to say that this architectural work is technically demanding and it requires savvy leadership. In the design of data centers, we can measure cooling capacity, electrical requirements, numbers of fire extinguishers, and so on. For architecture, we rely more on expert judgment and effective process leadership by empowered management. That's because there is not an abundance of quantitative measures and criteria to guide the process, though my colleagues at the Software Engineering Institute are advancing the frontier of more rigorous architectural practice.

(6) *Sixth, create a business and decision model that accounts for costs, risks, and benefits. This model provides a framework for expert judgment and effective process leadership.*

The goal is to have a decision framework to assess the value created by the engineering steps within the architectural master plan. At the architectural level, exact accountings and cost estimates may be less important than solid expert judgment regarding costs, risks, and benefits. Indeed, one of the challenges in IT is that it is sometimes easier to construct a seemingly strong quantitative case for a series of dangerous incremental changes than it is to make a quantitative case for a small disruptive change that moves us into a much more efficient and capable mode of operation.

A well-crafted business process would support incremental and iterative development in a way that any risk issues associated with the overall architecture and master plan can be addressed and discharged as early as possible. A range of techniques can support this, such as prototyping, side-by-side experiments, modeling and simulation. The point is that when a feature of the master plan has some associated uncertainty, experiments can be done at low cost to rapidly converge on a good decision, which can then be implemented with greater confidence. This is standard industry practice.

Summary

The most important features of my proposal are the emphasis on overall system architecture and the necessity of senior agency leadership to lead the process. It is important for senior agency leadership to participate in the master planning process and fully commit to the outcome.

This recommendation to focus on architecture is in harmony with the recommendations both of the NRC panel chaired by Professor Lee Osterweil and the Future Systems Technology Advisory Panel (FSTAP).

The Social Security Administration has an extraordinary and critical role in the lives of all Americans. I hope that the ideas I share today will be of use in strengthening that essential institution.