

Government tech projects fail by default. It doesn't have to be this way.

Author: Mark Lerner

During the early months of the pandemic, unemployment insurance systems across the country buckled under the weight of the rush of applicants, bringing tech failures to light in the most critical moment of many peoples' lives. Of course, government tech failures are not new. There's a report from the Standish Group that shows that government tech projects over \$6M only succeed 13% of the time. Examples of failing government tech projects are common knowledge and regularly make headlines, such as the famous Healthcare.gov incident, or the Hawaii missile alert system, or even the very recent UK COVID-19 data tracking failure. This has been going on for many, many years. The trend is so common that it's crossed over into a trope. It's safe to say that government tech projects fail by default.

Many researchers, practitioners, and observers have outlined reasons why these projects fail: mismanagement, lack of budget, lack of executive support, inefficient talent pipelines, inadequate design practices, and so on. When government projects fail and technology is involved, the same ingredients ultimately come up in postmortems. Even though we've come up with new ways to handle contracts with software vendors, to manage user research within the confines of government laws and policies, and to reduce risk in budget processes, we still find government tech projects failing to deliver value to the people that use these services and costing taxpayers hundreds of millions of dollars in the process.

So why is this the case? How is it that we understand what needs to change, yet we find ourselves at the wrong end of a government tech story time and again?

Why is it so hard to get things right?

From the outside, it may seem like some of the solutions are simple: adopt agile ways of working, bring in design practices, launch small products and improve them over time. These are exactly the principles and practices that forward-thinking public servants are trying to bring to their work. But, as many people that have tried to address this issue know, things are not so simple. At every level of

government, the various people involved are all trying to do their best – but are struck with out-dated requirements, restrictive policies, confining interpretations, and perverse incentive structures that not only keep them from doing the right thing, but push them towards working in ways we know will lead towards failure. In other words, the path of least resistance is the path towards failure.

Take, for example, the federal acquisition and procurement process. Government tends to contract out to external vendors for software licenses or development services far more often than it develops software in-house. These contracts balloon up in size, with \$100M+ software contracts being fairly commonplace. In many contract solicitations, the Request for Proposal is hundreds of pages long, with lengthy and very explicit requirements. Many federal agencies are, at this very moment, locked into several multi-year contracts that prevent them from changing to different vendors.

Of course, these problems have their own root causes. Contracting Officers tasked with defining these contracts and RFPs and solicitations are extremely experienced with the Federal Acquisition Regulation, but not with software development processes. Contracts can be so difficult and arduous to manage and produce that contracting offices try to do as few of them as possible, to minimize the pain of the process. Even when the software vendors and contractors know there are better ways of working, they're limited to what is written in the contract.

This is only one of many reasons why it's challenging to work in modern and agile ways. Congressional budget processes, duplicative committees on investment control and security and enterprise architecture, reporting requirements to agency and federal leadership, approval processes via the Paperwork Reduction Act or to gain an Authority to Operate – all of these factors weave together to create a strong network of incentives that pushes well-meaning public servants back into dysfunctional ways of working. On top of it all, when large government failures occur, the standard response is to spend more money and add more bureaucracy, rather than reevaluate why the existing bureaucracy hasn't prevented these failures in the first place.

How do we break this cycle?

To truly change how the government builds services in the modern era, we have to change these governance mechanisms. This isn't about adding another layer of oversight on top of the already too heavy machine. Instead, we need to carefully look at each of the nodes in this network – from performance management, to procurement, to intra-agency and inter-agency oversight bodies, to government-wide policies and regulations, and all the way up to Congressional budget processes and laws. Each of these plays a critical role in keeping the status quo as it is, and each has improvements that need to be made to clear the path for success.

The goal of this work isn't just to reduce failure, but to make the default government service one that works for all people that need it. This kind of work is somewhat removed from direct service delivery, but paves the way for better government services to exist.

My research here at the Technology and Public Purpose project focuses on these governance structures at the federal level. The first step is the broadest step – identify these governance mechanisms, and clearly show how they push well-meaning civil servants to build services that don't work for those that need them. From there, I'll begin diving deeper into tangible ways we can make changes in governance to not only remove blockers, but to actively encourage projects to operate in the ways we know lead to success.

Do you have experience managing government IT governance? Have you been at the whims of countless review committees? Have you tried to bring design and agile practices into your organization, only to be pushed back towards long-term waterfall software practices?

Send me an email at mlerner@hks.harvard.edu – I'd love to hear your story. Together, we can work to build a better government.