TESTIMONY OF JENNIFER PAHLKA, FOUNDER OF CODE FOR AMERICA AND
CO-FOUNDER OF U.S. DIGITAL RESPONSE
BEFORE THE COMMITTEE ON THE BUDGET, U.S. HOUSE OF REPRESENTATIVES
ON SOFTWARE UPDATE REQUIRED: COVID-19 EXPOSES NEED FOR FEDERAL
INVESTMENTS IN TECHNOLOGY
JULY 16, 2020

Chair Yarmuth, Ranking Member Womack and members of the Committee, I appreciate you inviting me here today to speak on this critical topic.

I have spent the past 10 years working with federal, state and local governments on their transition to effective delivery of services through technology and design as the founder and executive director of Code for America and also as the U.S. Deputy Chief Technology Officer. I had gotten used to how things work. In March of this year, when it became clear that COVID-19 was going to stretch the capacities of state and local government, I and several others started recruiting skilled tech professionals to work alongside government officials in an effort that became U.S. Digital Response, which has now helped over 60 state and local governments respond to the crisis at "the speed of need." This has once again given me the opportunity to see government technology and service delivery through the eyes of the uninitiated, and the question I am asked over and over again is, "Why does it work that way?"

U.S. Digital Response's tech volunteers ask the question out of genuine confusion and concern when they see how our government systems work. Many of our volunteers have spent their careers building the digital experiences we rely on today when we order and pay for goods and services, communicate with our friends and colleagues, or research a new topic. People use their products and processes every day with confidence. These are not whiz kids enamored with every latest gadget and fad. They are professionals who see technology as a way to serve vast numbers of people, humanely and effectively. They know that the trick with tech—and, coincidently, for a government of the people—is to get it to work for real human beings in all their glorious diversity. Agile, scalable, human-centered technology is important because it allows you to be responsive to changing conditions and human needs.

Changing conditions and acute human needs are why we are here today. A catastrophic event threatens the lives and livelihoods of millions and the only institutions with the ability to respond at this massive scale are governments. And yet we find that scaling up to meet the moment is exactly what American government at every level is struggling to do. Up to 15 million eligible people have not received their stimulus checks and, sadly, it's the least economically stable among us who are mostly likely to have been missed.[1] Families in need across the country are still waiting and wondering about their unemployment checks. Congress has pressed the gas

---

[1] https://www.newsweek.com/many-15-million-americans-could-still-missing-stimulus-checks-1512830

pedal all the way down, but the engine isn't revving the way we need it to and it looks like the hill we're climbing is only getting steeper. Why is this and what can be done?

First, I hope everyone here can relate to the stresses that government systems come under when asked to do 10 times the work they did before our current crisis. Yes, we can give them more resources, but new staffers must be trained, policies and procedures clarified and implemented, and these are not exactly normal working conditions; COVID-19 affects the offices assigned to dealing with the crisis in disruptive and challenging ways. We expect—and should expect—the human and procedural part of government systems to take some time to scale.

The part of these systems we expect to flex and scale quickly and efficiently is the technology, because that's what they do when we use them in our daily lives. But the technology most of our governments are using to help people access critical services is sorely outdated and lacks the capacity to do just that, for a number of reasons. We must invest in modernizing the technology that runs our services, but I am deeply concerned that the urgency of the moment will cause us to forget that **we must also change how we make these investments**. Now more than ever, we cannot afford to pour time, attention and enormous sums of money into a process for building and buying software that has not worked for decades.

Let's take unemployment insurance benefits. The state systems that deliver this service rely on a hodgepodge of legacy systems onto which websites have been bolted. As has been widely reported, at their core, many of them still use a mainframe system programmed in COBOL. COBOL is a programming language dating back to the 1970s, which means that it's doing a great job standing the test of time. If you're still driving a car from the 1970s, that means it was built to last and it's a classic! But you can't expect that car to suddenly get the gas mileage of a modern hybrid or electric car. At best, it's going to perform as it was designed to do decades ago.

We're asking that 1970s car to do quite a few new tricks today.

We're asking it to be **agile**: the pre-COVID unemployment system had a single application form, for conventional unemployment insurance, and a weekly certification form. Now there are three applications for three different unemployment programs, with dependencies between each, and all those changes must be reflected in the digital code that runs the system. More modern programming languages are designed for greater speed and flexibility and would make that task easier. But the bigger barriers to the quick adaptations we need are the shortage of COBOL programmers in the market and the fact that the code that runs these systems has evolved in archaeological layers over decades; as technology and policy have changed, they've been modified and hacked here and there to the point that a precious few long-standing employees know how these systems work. In some cases, no one at the Department of Labor or the vendor who provides the system actually knows how the system works anymore. It is very hard for systems to adapt quickly under these conditions.

We're asking our systems to **scale**: in one state, from about 6,000 inquiries a week before the pandemic to about 50,000 per week since, almost nine times the volume. If it were just a matter of processing checks, these legacy systems would perform reasonably well. But in one state that U.S. Digital Response worked with, only 33% to 50% of applications were automatically accepted or denied; up to 67% of applications required review and determination of next steps by a staff member. This large number of exceptions means not only that the load on the system is far greater, but it also illustrates how much the speed and quality of service delivery is a function of policy and process as much as technology. This is why we advocate for hybrid teams that include people who touch every aspect of the technology, design, policy, process and compliance working together to get the results we want. Technology, especially software that runs in the cloud, is uniquely capable of scaling up on demand, but when the process requires manual intervention that benefit disappears.

And we're asking it to **work for people**, for the questions and the process to be clear so applicants know how to answer them accurately without assistance, for the people who administer the program to be empowered to make decisions that get the benefit to those in need as fast as possible. That's hard to do when, as the director of Michigan's Unemployment Insurance Agency told *The New York Times*, these systems were "built to assume that you're guilty and make you prove that you're innocent," and partly as a consequence, pre-COVID-19, only one-in-four unemployed people in Michigan received benefits.[2]

Neither the technology nor the policies were designed for any of these things, and we have known this for decades. Ten years ago, the National Association of State Workforce Agencies (NASWA) stated clearly that "most state IT systems cannot efficiently handle today's demands" and detailed the skyrocketing costs that states pay: nine out of 10 states reported support costs that rapidly escalate every year.[3] Many states joined consortia to contract for new systems together, in the hopes of reducing the cost to each. Some of these projects have been implemented in the states for which they were procured; others are somewhere along in 10-, 12-, even 15-year procurement and development cycles. Collectively, they've spent billions of taxpayer dollars. And yet, when people started applying for unemployment at levels not seen since the Great Depression, even the states that had supposedly modernized struggled to respond.

Some say these projects were underfunded or that we wouldn't be in this position if modernization had started sooner. The reality is that neither of those would have helped much. As long as these projects follow the basic recipe of conventional government procurement and development, the result will be the same overscoped requirements documents, the same bidding rules that ensure the contract will go to the same vendors, the same disconnect from the policy teams that prohibits collaborative problem solving, and ultimately the same outcome.

---

[2] https://www.nytimes.com/2020/04/30/upshot/unemployment-state-restrictions-pandemic.html
[3] http://www.itsc.org/itsc%20public%20library/NationalViewUI_IT%20Systems.pdf

Our mistake has been believing that we're funding "modernization" through projects that take over a decade to produce. We know technology moves too fast for something designed 10 years ago to be useful or cost efficient. Would you pay top dollar for a 12-year-old cellphone? More importantly, our world moves too fast—policies change, habits change and, as we're seeing acutely today, needs change. But our bigger mistake has been thinking we're funding modernization when we hire a vendor to take a spaghetti bowl of policy and process that's accumulated over decades and simply recreate it on slightly less outdated technology platforms without rethinking the design of the service itself. To truly modernize our services, we must not only use more current technology; **we must prioritize agility and human-centered design in both the development of our services and the services themselves.**

To get government tech right, we of course need to be able to procure more modern technology platforms. But that will be insufficient if we don't also do three things that support **agility and human-centered design.**

- The first is to break down the silos between policy, technology and other disciplines. Technology can't speed a process in which most cases must be handled manually, as I described above in the case of unemployment benefits under the CARES Act. A similar problem is that many states require applicants for Pandemic Unemployment Assistance (PUA) to apply for regular unemployment first, wait to receive their rejection, and only then apply for PUA. Tech, operations, policy and compliance staff must work together to solve these problems, and agile development models allow for this collaboration in ways that legacy models do not. We must even have digital professionals at the table when we craft policy; understanding how the service will be delivered is critical to getting the outcomes the policy seeks, especially now, as we face greater and greater needs and limited delivery capabilities. As the former head of the White House Domestic Policy Council Cecilia Muñoz has said, "Policy leaders must learn the skills of human-centered design, and technology must have a seat at the strategy table."

- The second is to encourage rapid prototyping and continuous development. Our legacy process involves a requirements gathering period that can take many years, followed by the development of a Request for Proposal that can be thousands of pages long, lengthy contracting and development periods, and then a move into what's called sustainment. This process may work for constructing buildings, but it's simply not how good software comes to life. It is better, faster and cheaper when interdisciplinary teams start small, build iteratively, work closely with the users of the software all the way through, and continuously update and improve the application.

- The third is to demand that all services provide real-time data about their usage and that human beings are assigned to looking at that data to understand what's working, what's not working and what can be done about it. When Code for America started working to decrease the participation gap in Supplemental Nutrition Assistance (SNAP) in California, our team found that the program leadership had very little insight into the

reasons people tried to apply and couldn't, or applied but couldn't make it through the burdensome process despite being eligible. It wasn't that they didn't care; the systems they'd been given to manage eligibility and enrollment simply didn't provide that data, and what data they did get was usually months, if not years, old by the time they got it. Creating an online application that was simpler and easier to use had huge benefits for the people applying, but an equally important benefit was that the system was instrumented to allow decision-makers to see in near real-time where users got stuck and begin to fix those issues. This access to real-time data is part of what's needed as we deal with today's crisis.

These agile, human-centered models for developing government software systems work. When public servants are allowed to use them, these models have a far higher success rate, which is not difficult given that the legacy model has been estimated to fail 94% of the time. These new models reduce risk and the projects that use them cost less and deliver results faster. They are first and foremost designed around meeting the needs of the users of the system (both the public servants who administer these programs and the American public who use them), and secondarily around meeting the significant burdens of compliance that have been placed on them by this body and others over the years.

These models are currently in use at all levels of government, at places like the United States Digital Service, GSA's 18F unit, the U.S. Air Force's Kessel Run, the State of New Jersey's Office of Innovation and the Colorado Digital Service. This model was given the official stamp of approval by the Office of Management and Budget under Federal CIO Steve Van Roekel in the form of the Digital Services Playbook[4] after a group of people who practice this model helped rescue the embattled healthcare.gov site. This model is responsible for subsequent successes at the Department of Veterans Affairs (the new va.gov), the Federal Elections Commission, the Centers for Medicare and Medicaid Services, U.S. Citizenship and Immigration and dozens of other places.

It's why the Department of Defense turned to the Defense Digital Service when it needed a COVID-19 symptom tracking tool within a week[5]. Leadership knew they could deliver based on their track record of speed, quality and cost; in an environment where it routinely takes 12-18 months **after the project is complete** to get an authority to operate (ATO), DDS had just built a software application for a logistics workflow within 90 days and had it approved to launch in just over two weeks. These are not private sector practices that we naively hope will work in a government context; they are grounded in proven fundamental principles and adapted to work with our existing law, policy and regulation. The good news is that Congress doesn't need to pass a law to make these practices legal.

---

[4] https://playbook.cio.gov/
[5] https://mysymptoms.mil/

However, despite the documented successes of this agile model, the vast majority of government technology projects still do things the old way. Why? For one, because while this isn't complicated (in fact, this way of doing things is much simpler) it is still very hard to do. It requires the support of leadership across silos, because technology, design, procurement and policy must work together as one team for this model to work. It requires the support of leadership because it looks very different in practice from how traditional acquisition occurs, and because of that, it makes stakeholders of all stripes nervous, and even threatened. Leadership at all levels and across many domains must be there to protect digital teams using this model from constant attacks.

It saddens me to report that when I ask defenders of the status quo why they won't support an agile model for a project, or even why they are actively undermining one to revert it to the old model, the most common answer is that Congress won't let them. I don't believe this to be true, but there is certainly a disconnect and an enormous fear of being seen to have broken protocol, even when that protocol is "how it's always been done" rather than defined by law. For example, it is perfectly legal to use Other Transaction Authorities (OTA) to support an agile procurement, but many contracting officers will refuse to do so out of fear that someone—anyone!—will say it was improperly used. The safer thing to do is to take the longer, more complex path and justify it by saying that any deviation in the model could result in being called before Congress. And no one wants to be called before Congress.

Congress can no longer be used as an excuse for holding back progress in our digital capabilities when the American people need government services to work better and faster than ever. To fix this, Congress will have to be more than a checkbook; this body will have to become a staunch ally of hybrid tech-policy teams who practice agile development and user-centered design, whether they exist in the USDS, GSA, federal agencies, the military, state agencies responsible for services, at new efforts like the State Software Collaborative at Georgetown's Beeck Center, or, even at vendors. It will have to visibly demonstrate support for agile projects, teams and practices, and back them even when they experience setbacks; small, visible failures early on are part of the process of learning, but can be weaponized by protectors of the status quo if those in charge allow it. In addition to rewarding the use of Other Transaction Authorities, Congress should double down on the tests it has currently authorized at the Department of Defense to let nine programs break free from "color of money" strictures and ensure no other agencies are forced to budget for software in ways that hold back critical progress.

Congress will have to also be a watchdog, but a different kind of watchdog. As we move from outdated development practices, Congress, Inspectors General, the Government Accountability Office (GAO) and others will have to reinvent oversight, changing assumptions (such as that more money is better for a project—we now know that something akin to the opposite is true) resetting expectations (expect working software much sooner and to fund continual iterative development rather than distinct build and sustain phases) and asking different questions.

Congress will have to stand up to the companies who benefit most from the status quo, as they have invested a lot in playing the game with today's rules. You will be told by lobbyists and others that terrible things will happen if we invest in governments' digital capacities and do things differently. The reality is that these companies will be fine. If we change the rules of the game, a few contracts will go to new players and some contracts will be smaller, but there will be more of them because there's so much work to be done. You need only to look around to see the needs! And the companies will adapt to the new rules. They'll be fine in this new world, but the American people will not be fine if we don't make these changes.

Most importantly, Congress needs to support different ways of building and buying software, and expanding government's core competency in digital delivery. We will always hire vendors but, today, we often don't even know what we really need when we contract. For example, states are still struggling to deliver on the CARES Act provided unemployment insurance, as applicants wait weeks and even months for their application to be processed. More legislation is on the way to dictate additional relief, current thinking would suggest more money for each of our 50 states to support the software systems that deliver unemployment and other benefits. Some of that funding is definitely needed, but it will cost much more if we ignore higher leverage opportunities. For example, while federal legislation dictates eligibility for each program, every state must figure out how to determine whether a given applicant is in fact eligible, which requires systems to verify their identity, validate their reported income, check on whether they've applied for or received benefits in another state and screen for fraud. This is not only a technology and data challenge that could be solved once for all states, it's also the single biggest cause of delay in receiving benefits, as each state bears the burden of proof. A state Department of Labor employee operates out of fear—much like the procurement officer who won't use an OTA—that someone will say that she has approved someone who was in fact ineligible. A central service for eligibility checks that also gives each state safe harbor to award benefits without repercussions if they use it would cut out weeks of delays in millions of cases around the country, much as E-Verify works at the federal level to allow employers to hire workers. Money is one resource; tech and service delivery insight and expertise is needed to surface and implement these kinds of interventions. Increasing funding for digital service teams like USDS will save millions, even billions of dollars in the long run, but more importantly, get help quickly where it's needed.

Change will not happen without leadership and political will. More funding that flows into the legacy model will get us more of the same. But if we recognize the need for real change, systemic change in which many stakeholders will need to play a part, and do the much harder work of building government's capacity for digital service delivery, protecting our nation's innovators instead of constantly letting them be crushed by a culture of fear, we can get the American people what they need. What Congress has already recognized they need—through services that are in fact, though you've been told you can never have all three—better, faster AND cheaper. Services that are flexible, scalable and designed to work for people. To do that, to use a metaphor beloved by my former boss, U.S. Chief Technology Officer Todd Park, you'll need to stop arming the empire and start arming the rebels. Stop pouring hundreds of millions of

dollars into projects everyone knows will fail. Fund the United States Digital Service and state digital services who can help others across the government ecosystem safely and successfully implement these practices. We have the know-how to make government effective and responsive. We need Congress's air cover to run the new playbook, so government gets ahead of needs, not woefully far behind.

A common proverb tells us the best time to plant a tree was 20 years ago, and the second best time is now. It is true that the best time to modernize government services was 20 years ago, because change of the kind we need will take time. We can and must move much faster in the development of digital services in order to succeed, but changing behavior across the vast landscape that is government is much harder than changing a rule. Which is why we must do it now.

Indeed, the tree is planted. Today, we are blessed to have some of the best digital professionals in the country already working for government at places like U.S. Digital Service. We have hundreds, even thousands, more waking up to the impact they can have by working with government through efforts like U.S. Digital Response, Code for America and others. More and more dedicated, passionate public servants want to truly serve the public by following the digital services model instead of the broken model that doesn't work. More and more examples of success exist to inspire them. But this tree is still a tiny sapling in a giant forest. It needs water and sunlight. It can only get them if our leadership is willing to clear some space.