



## How the U.S. Department of the Treasury Pay.Gov Team Went Agile

### SUMMARY

This case study provides the following information:

- How the US Treasury and its partners implemented Agile
- The results of the implementation on increasing the quality software products
- How the implementation increased the efficiency of shipping products to market
- The change in workflow that allowed for quick shifts in priorities
- The resulting cost savings or collecting funds electronically

Level of Acquisition: **Expert**

By JoDee DeVillier, Shawn Leitner, and  
Tim McFadden

U.S. Department of Treasury, Pay.gov Team

The US Treasury, Federal Reserve Bank of Cleveland, is responsible for the Pay.gov system. This system allows customers to make electronic payments to government. But after a series of product issues, the team decided to try a new approach to building and maintaining all of the products that together, created the customer experience for Pay.gov. This involved using a contract that supported Agile Project Management. The following is what happened once that contract was in place.

## Background

One of US Treasury's systems, Pay.gov, provides services to 1,000+ federal government business lines to allow customers to pay the government electronically, collecting 160 million transactions for \$160 billion annually. Pay.gov has gone through numerous iterations since its inception in 2000, including three development teams in three different organizations. In 2011 Pay.gov was starting to grow rapidly as electronic collections in the government started to take off. More features were being added to Pay.gov at a rapid pace. Priorities were shifting. High-profile agencies were joining at a rapid pace. Customer behavior was changing. Expectations were rising.

We had to explore various options that would solve these very complex issues. This case study tells the story.

## Lessons Learned

Here are some examples of our lessons that we learned.

1. Pick the right project to start your Agile journey – We had a very large and visible project upcoming that could not fail and at our trajectory, we were trending towards failure. In an effort to avoid that fate, we decided to start our transition to Agile with this project. While the scope was pretty well-defined, the project required re-engineering the most critical portion of our system. With the benefit of hindsight, deciding to change methodologies right before we started one of the largest and most ambitious undertakings we had, regardless of software methodology, was a lesson learned.
2. Cultivate all relationships, not just developers – The switch to Agile was a grass-roots effort, coming from the development team. This made the transition for the development teams easier (but not easy), changing the culture and mindset of the entire organization, from testers to the Product team to senior management and other stakeholders, was even underestimated and even more difficult.
3. To the victor go the spoils – During our transition to Agile, we started seeing the fruits of our labor. Software quality was up, we were accomplishing more scope, and our releases were on-time, so agencies wanted more features. This required scaling our team significantly before we even perfected our implementation of Agile. We learned a

few hard lessons here, and we eventually brought in a consultant to check in and see how we were doing.

4. How long it takes to transition to Agile – When we embarked on our Agile journey, we knew it was going to be a lengthy transition, we just didn't realize how long. You will see in the subsequent sections that it was a multi-year effort. And in true Agile fashion, there is no end to continuous improvement, so we are continually looking at opportunities to improve.

## The Problem

The problem soon became very evident. Our features did not meet customer expectations. Our time-to-market was slow. We could not adapt quickly to scope changes. Our processes were exclusive. The demand for features was intense, necessitating significant scaling of a small operation. But most of all, our software quality was low, resulting in significant regressions, customer frustration, disruption, and project volatility.

## Finding A Solution

Once we acknowledged the problem, we took action to implement Agile. As stated earlier, we knew it was going to be a lengthy transition, and that would require different aspects to make it successful.

### 2012

1. Story writing training – Switching to Agile required us to create requirements differently, focusing on the who the customer is, what they want to achieve, and why they need us to build them a product. User stories are written in a specific context to help all parties remember the who, what, and why:  
As a <type of user>,  
I want <some goal or objective>  
so that <benefit, value>
2. Scrum coach – We chose to implement Scrum which is a set of practices that allow an organization to implement Agile. This required a completely different process for all parties so we hired a coach to help us implement Scrum.
3. Learn from others – We wanted to learn how other companies implement Agile, so we visited with a large insurance company and large data analytics firm to see how Agile works for them.

### 2013

1. Scrum Master training – The Scrum Master role is key in an Agile and Scrum environment, so three managers went to Scrum Master training
2. Cucumber and Gherkin training – The key to any developed feature is to ensure that not only do the new features meets customer expectations and is of high-quality, but that it

doesn't break any existing features. We chose to do Cucumber and Gherkin to implement a suite of automated tests to ensure that we could meet both.

## 2014

1. Test-driven development (TDD) training – The development team participated in training to learn how to think like a tester before developing new features.
2. Agile conference – The development, test, and Product teams sent representatives to an Agile conference to understand how other organizations have implemented Agile and Scrum. We discussed practices, processes, tools, and organizational structures.

## 2015

1. Re-examine all aspects of our Software Development Lifecycle (SDLC) – We brought in a consultant to examine all aspects of our implementation of Agile. Did we scale properly? Are we aligning with Agile principles? Are all groups engaged? Are we delivering and communicating properly? Is our organization structured properly?

## 2016

1. Organizational restructure – The consultant, among other things, recommended changes to the organizational structure, specifically the Product and testing teams, and those were implemented.
2. Scrum Product Owner training – The findings from the consultant facilitated not only an organizational restructure of the Product team, but an expansion of its construct and influence. As a result, new members of the team needed Scrum Product Owner training.
3. User Experience (UX) Design – We realized that we were not spending enough time focusing on the user experience. Agile allowed us to create high-quality software features, but without a dedicated focus on the user's experience, the feature may not reach its full potential. We found that a strong Product Management discipline focuses on user experience with trained resources dedicated to and advocating for delightful user experiences.
4. Product Management strategy – Documenting and socializing the Product Management strategy, from roadmaps to requirements, from deliverables to escalation, from risk management to project statuses, provides everyone insight into the Product Management organization.
5. Product Management conference – The Product team sent representatives to a Product Management conference to understand how to scale and cultivate strong Product teams, and how they function at other high-performing organizations.

## Results

We expected to see positive results from our transition to Agile, but the results have been even better than we expected.

1. Accomplish more scope – We are able to accomplish more scope because we have the right people involved in the right processes at the right times. For example, in our April 2017 release, we added two fraud prevention items to our scope after the release began. We were able to complete this effort without changing the implementation date, incurring significant risk, nor altering any other scope.
2. Features meet customer expectations – We only work on products that have customers ready to use them, and involve them throughout the planning, development, and testing phases. This ensures we develop products that meet customer expectations, lowers risk, increases customer satisfaction, and lessens volatility on the project teams. For example, we developed a billing solution in our April 2017 release that exposed some assumptions and communication gaps, but because we had a close partnership with the agency throughout the development process, and we were able to quickly correct our course to still satisfy the business need without causing undue risk.
3. Embrace Change – We are able to absorb and embrace requirement and design changes. We all know that changes occur, and having an inclusive prioritization process helps all stakeholders understand the change and why. For example, in our June 2016 release, after development had already started, we encountered an issue with our design of our integration with a third-party settlement provider. We quickly pivoted and proposed a new design that aligned better with the program strategy and lowered the risk of the implementation, yet still met the business need and provided a good user experience.
4. Software quality – We switched to Agile primarily because our software quality was low. Other factors were also present, but the primary driver of our switch to Agile was low software quality. Of the 14 releases we have developed since switching to Agile, one release required a patch release, and one has missed their deployment date.

## Conclusion

The transition to Agile was turbulent and challenged our organization and partners in many ways. This challenge was driven by numerous factors, most of which took some heavy lifting by many parties. Ultimately, the transition to Agile has allowed us to accomplish more, communicate better, produce products that meet or exceed customer expectations, and adjust to the changing times better than ever.