Product Validation Summary

Better Document Submission

February 2025

Executive Summary

Document Submission Overview

Submitting documents is a critical part of almost all benefits applications. Almost a third of Americans with incomes less than $30,000 rely on their phone for internet access, thus mobile-friendly applications are critical to meeting beneficiaries where they are.

Since 2019, the adoption of mobile-responsive benefits applications has increased by 25%, improving document submission for beneficiaries. However, processing these documents remains a burden for administrators.

Once digitally submitted, documents received by an agency must be categorized, reviewed for legibility, and converted into machine-readable formats for routing to internal systems that support decision-making. While some agencies have adopted mobile-first tools to enhance the front-end application experience, the back-end process for administrators remains largely manual. As a result, mobile uploads have unintentionally increased the burden on administrators, who now must process a higher volume of documents.

Processing documents and extracting critical information to make application decisions faster and more efficiently continues to be a persistent pain point for administrators.

Our proposed solution pathways explore **how we might reduce manual document processing – streamlining submission and data extraction – so administrators can focus on making rapid decisions.**

Recommendation

Move “Better Document Submission” to the next phase of development.

We strongly urge investment in one of the two prototyping pathways for a “Better Document Submission” tool. Our proposed solution is well-positioned to meet TTS's criteria for success, delivering impactful, scalable improvements to document processing across government agencies.

Desirable: Nearly all our interviews with potential customers (state benefits agencies), delivery partners in the document solution space, as well as benefits systems experts, have pointed to a need for better document processing tools for public sector agencies.

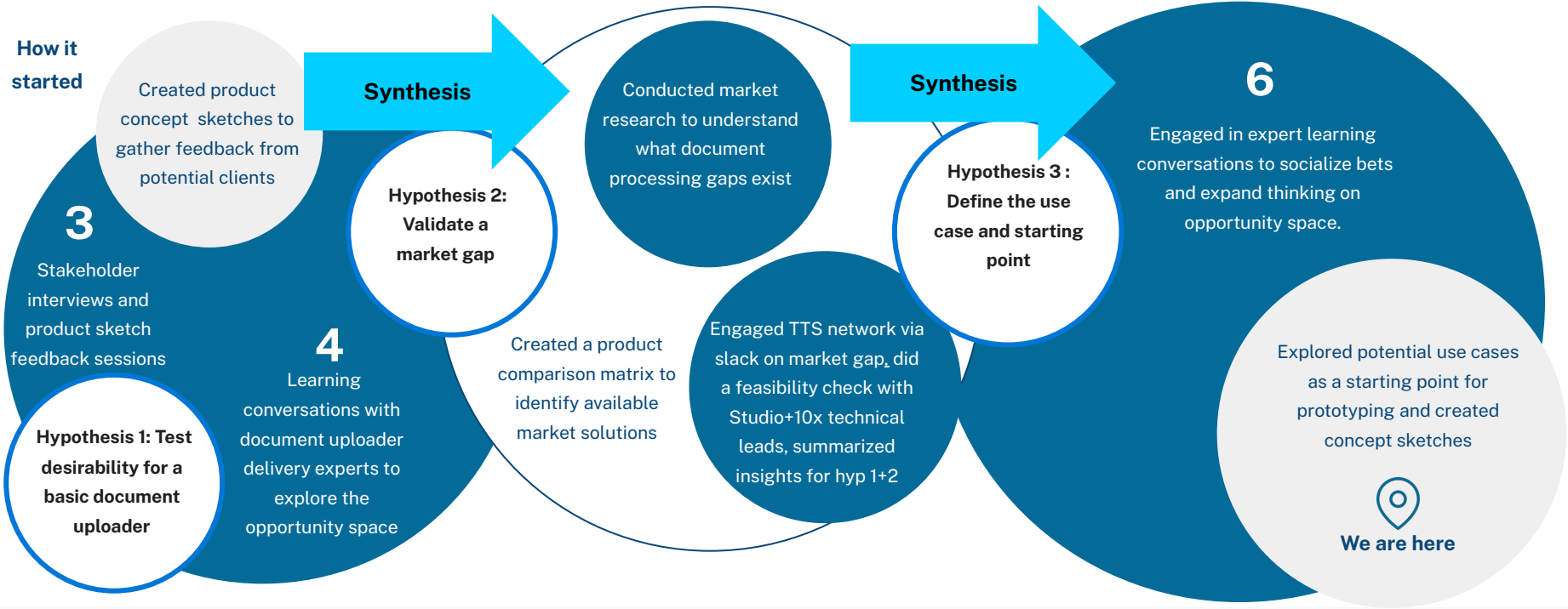
Viable: Reducing manual administrative burden is essential for agencies operating with tight budgets and limited staff. Business departments are willing to invest in tools that enhance efficiency and have historically trusted federal solutions.

Feasible: Our proposed solution aligns with existing commercial technologies. While OCR and similar technologies are not new, recent advancements make them a strong foundation for testing and future growth.

Leverage TTS: A “Better Document Submission” tool has wide applicability, any agency that collects information from its customers is a potential use case – an ideal case for TTS’s ability to work across government to make it easier and cheaper for people with the same challenges to use the same tools.

Research

Our research journey



Details on learning conversations and "Document Submission" project outputs

Product Exploration

We once thought: The simplest product we could build that provides value is a mobile-first document submission tool that delivers documents to an inbox.

We have since learned: The pain point that came up persistently from state agencies, document uploader delivery experts, and benefits experts, over document uploading, was document processing. Further, there is an existing gap between agencies' document processing needs and what the market is offering that warrants a TTS provided tool.

Assumptions we investigated

H1

The simplest product we could build that provides value is a mobile-first document submission tool that delivers documents to an inbox.

Not validated

H2

There is an existing gap between agencies' document processing needs and what the market is offering that warrants a TTS provided tool.

Validated

H3

The simplest product we could build that provides value is a {mobile-first} submission tool that is able to process images of documents into usable data and can be implemented as a component alongside existing workflows.

Next Step

More information about our hypotheses here.

Why start here?

Assumption: Submitting documents as part of a benefits application is a significant pain point for applicants. Many state benefits systems still rely on paper and fax-based submissions, creating a tedious, time-consuming and costly process for both beneficiaries and administrators..

Secondary Research

Failure to submit documents is a critical fail point in the benefits application process:

Missing documentation is one of the primary reasons for benefit application denials. By building a mobile responsive document uploader for CalFresh, CFA was able to increase the number of verification documents submitted by 60% and found a significant positive correlation between documents submitted and application approval rates.

Mobile solutions are growing but are not where they need to be.

As of 2024 69%, (25% increase since 2019) of online benefits applications were mobile accessible. But, in many cases, this does not translate to a fully mobile experience for an application process. Beneficiaries may still be asked to submit paper documents or documents as .PDFs or as a .doc/x when submitting photos of documents (i.e. as .jpeg, .png, .HEIC) is the most intuitive way to send a document using a phone.

Document uploader technology exists, but adoption lags.

During the COVID-19 pandemic the DOL worked with several state UI programs to understand their document challenges and provide best practices and guidance. They found that before the pandemic, many state UI programs lacked document uploaders that would enable claimants to digitally send verification documents. States that invested in document uploaders, though, said it was a game changer.

Identifying the right problem

Hypothesis 1: The simplest product we could build that provides value is a mobile-first document submission tool that delivers documents to an inbox.

Finding: The most pressing problem for agencies is document processing over document submission. Agencies recognize that mobile-first tooling is the way to meet their users where they are. But, backend document processing burden continues to be pain point for administrators, implementing mobile uploaders has not addressed this problem and in some cases has made it worse.

Key Learnings

Mobile-first shifts the burden

As agencies update their customer facing solutions, mobile friendly is the key requirement. However, the capability shifts burden from claimant to administrator, increasing manual reviews on documents received which extends benefit claim timelines.

Manual processing slows decisions

Manual interventions, a key processing step once a document has been uploaded, hinder rapid decision making for claims which significantly impacts the claimant experience. Manual interventions include extracting text/numbers from a document and inputting into a DMS, doing calculations, and verifying a document is valid or filled out correctly.

System compatibility is essential

Any component we build must be modular enough to implement alongside agencies' existing system as document handling is a core function of application processing.

Transitioning to Hypothesis #2

Open questions from Hypothesis #1

- What market solutions exist for document processing? Why aren't states using them?
- Is there a need for document uploaders amongst certain state agencies but we have not identified the target market yet?
- What are the specific document processing needs of state agencies?
- If processing is the issue, will we still need to build an uploader?

Towards Hypothesis #2

We know agencies that have built document uploaders are shifting their focus towards ways they might automate processing workflows, but large gaps remain.

Our next inquiry focused on: *Where do these document processing gaps exist, and why isn't the market addressing them?*

Understanding the market gap

Hypothesis 2: There is an existing gap between agencies' document processing needs and what the market is offering that warrants a TTS provided tool.

Finding: Yes, this gap exists. The way most government agencies approach technology modernization prioritizes large system-wide changes over smaller upgrades to tooling. Given the sporadic nature of funding and the already dated technology, agencies prioritize urgent "on-fire" changes to efficiency improvements. Even when agencies attempt a modular approach to tooling, they find most off-the-shelf tools require external expertise to implement and do not quite meet their technical requirements.

Key Learnings

COTS solutions don't always reduce manual workflows

There is no uniform way agencies are approaching COTs solutioning and often times COTs do not reduce manual workflows.

The result is a patchwork of complex home grown systems being built for organizations with shared data needs.

Implementation requires specialized expertise

COTs are not easy to implement within agencies' diverse ecosystems and most agencies do not have the talent in-house to implement these or custom solutions. This increases dependence on customized vendor solutioning and external expertise.

Procurement favors large, complex projects

Agency approach to purchasing leans towards large complex multi-year engagements when modernizing IT often leading to technology that is outdated before it's live. Because most technology upgrades are often funded through grants, the effort to upgrade consideration favors large projects.

Transitioning to Hypothesis #3

Open questions from Hypothesis #2

- How do we find and target customers who are interested in modular technical systems?
- How do we encourage agencies to reach for modular technical solutions in a landscape that is dominated by large enterprise systems?
- How do we build something that could integrate into agency systems without requiring significant technical expertise or being disruptive?

Towards Hypothesis #3

We know the solution is a tool that helps increase efficiencies and reduce burden on administrators (i.e., improve the backend) as they try to reduce burden on beneficiaries (i.e., improve the frontend).

Our next inquiry focused on: *What does the smallest testable intervention look like?*

The shape of our solution

Hypothesis 3: The simplest product we could build that provides value is a mobile-first submission tool that is able to process images of documents into usable data and can be implemented as a component alongside existing workflows.

Finding*: We recommend prototyping a tool that is not primarily a document uploader. The riskiest piece of the puzzle to test is improving the processing of documents on the backend. We can test this without interacting with the frontend intake of documents and without building integrations with an existing document system.

**Findings still in progress, our assumptions have been validated in initial interviews but still need to be tested.*

A successful solution must reduce administrative burden and integrate with existing systems.

Any document solution we build must improve processing burden for administrators in order for it to be a worth their time, energy, and resources to implement.

Additionally, our solution must be implementable alongside agencies' existing systems as document handling is a core function of application processing – an ideal case for a modular components-based approach.

We have identified intervention points and received strong expert feedback

We identified potential intervention moments and created initial wireframes of a prototype but we still need to test it.

We test drove our solutioning concepts while conducting landscape outreach discussions with experts and received near universal positive feedback from experts – ***CfA, Beeck Center, Washington state IE&E team, New America, USDR.***

Income documents provide the strongest initial use case for testing

Income documents – pay stubs, W2s, 1099s – are an ideal initial use case for prototype testing as total income must be calculated from a summation of income numbers across all of one's income documents. There is clear value for automated data extraction. Additionally, all benefits applications require verifying income.

Advancing to Hypothesis #4: Testing OCR +Data Extraction Feasibility

Open questions from Hypothesis #3

- What are the OCR tools that are available to us for prototyping?
- What key data do administrative agencies want extracted from documents?
- How accurate do OCR outputs need to be to be for agencies to be able to use them? Is that feasible for us?
- Are there additional considerations for partnering with a program we have not yet considered that would shift our use case?

What's next for Hypothesis #4: Testing backend-data processing

We strongly believe that we can prototype a data processing tool, specifically extracting text from documents and transforming it into a machine readable data that can be input into existing Document Management systems, that provides value to administrators without building a front-end uploader or integrations but our next hypothesis will test that assumption.

Next on our roadmap: *Selecting a prototyping pathway and initial use case to validate whether our solution can meaningfully reduce administrative burden without requiring major system overhauls*

Overall Learnings

The core pain point for administrators is manual document processing.

Market gaps exist because of common government procurement issues.

A standalone document solution must be a modular component, able to be implemented alongside existing workflows with minimal technical expertise.

Uploading income documents requires temporary data storage, introducing potential PII risks.

You will find more details on market gap analysis here

Income documents are the most desirable starting place for testing. Prototype (1+2) may require temporarily storing documents.

Prototyping Options

Visualizing the problem End-to-End

A claimant submitting a document is only the beginning of the journey. The workflow that takes place after the document is submitted (processing) remains largely manual.

Claimant experience starts here

Claimant Enrollment: Either online, by paper or mobile device

Submitting Verification: Missing verification documents is a common reason for denials.

Document submission hand off to the agency for administrator processing happens here

Administrator processing experience starts here

Review: Determining if documents are legible and usesable.

Verify: Authenticate identification through external service (if available) and manually check other documents are real

Code: Extracting and inputting relevant information into other internal processing systems for further analysis.

Use: Utilizing relevant data from income documents to calculate and compile a total income for the claimant,

Administrators in the middle performing manual tasks to process documents

The result of these manual actions get us to a yes/no in determining a benefit claim

Prototype pathway 1: Automated data extraction

How might we get data out of a document into a machine readable format to reduce the manual burden of data entry for administrators?

Solution : An automated data extraction tool that converts PDFs and image files into machine-readable formats to reduce the manual burden of data entry for benefits administrators.

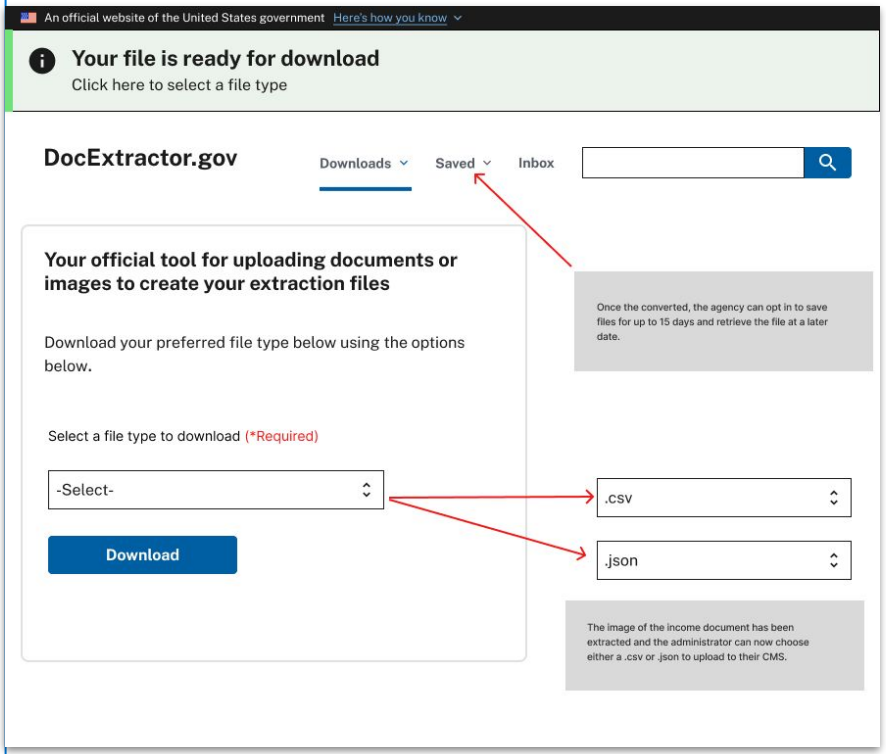
Role focus: Benefits Administrator

Benefits Administrator Task
When uploaded documents are received from benefits claimants, benefits administrators want the data to be extracted from the uploaded documents into their systems to facilitate benefits determinations and calculations.

How it works: Documents are converted into structured data.
1. The prototype takes in PDFs or image files (e.g., .JPEG, .PNG).
2. Extracted data is output as a .CSV or .JSON file.
3. The administrator uploads the extracted file into their existing system for processing.

Potential starting points to test

- **Document types:** Income documents (W2s, pay stubs, 1099, employer letter)
- **Programs:** UI, LIHEAP, SNAP, TANF



Administrator Intervention Moment

Reviewing: Determining if documents are legible and usesable.

Verify: Authenticating and and manually checking other documents

Code: Extracting and inputting data

Use: Synthesizing data to calculate and compile wage reports.

Check out the full flow here

Prototype Pathway 2: Mobile-friendly document submission

How might we support beneficiaries to submit their documents in an easier more digital way while ensuring we do not add processing burden to administrators?

Solution Mobile friendly document submission tool that converts docs to readable PDFs.

Role focus Benefits Claimant and Administrator

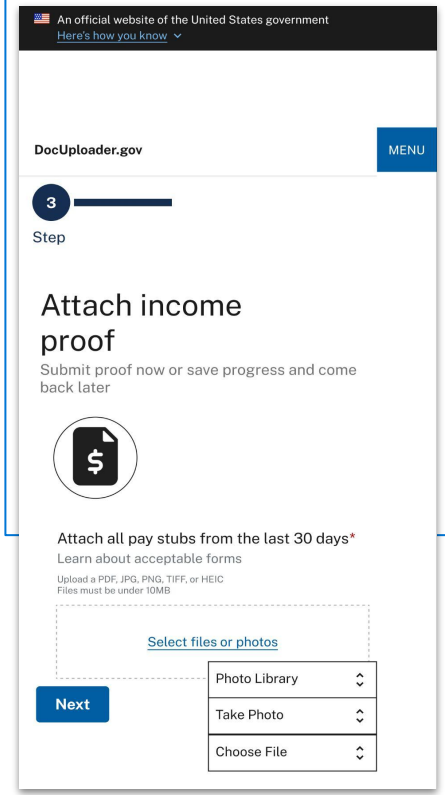
Beneficiary Task
When beneficiaries only have access to a phone (for internet use) they need a mobile friendly interface to upload documents proving their income and other eligibility to avoid denials.

Benefits Administrator Task
When uploaded documents are received from benefits claimants, benefits administrators want to ensure documents are legible and can be preserved for deeper analysis to make claim determinations.

How it works: Prototype converts images and PDFs into machine-readable formats.
1. The prototype accepts PDFs and image files from beneficiaries.
2. It converts them into a readable, searchable PDF where text can be recognized by software like Adobe.
3. If the document quality is too low, the prototype adjusts the image or rejects the input, prompting the user to resubmit a clearer version.

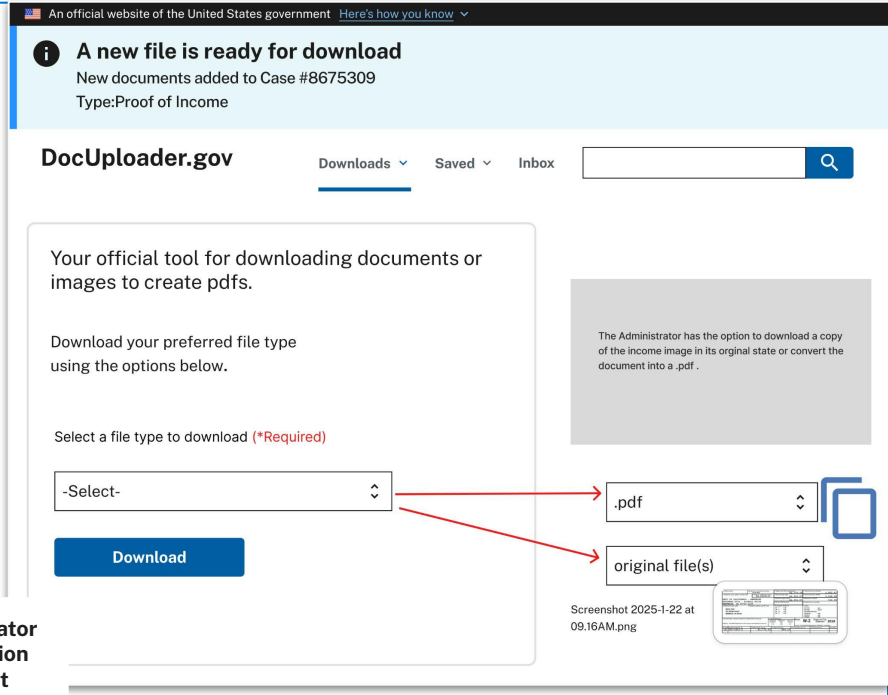
Potential starting points to test

- **Document types:**
  Income documents - *W2s, pay stubs, 1099*
  Enrollment in another benefit program - SNAP letter, Medicaid card
- **Programs:** UI, WIC, LIHEAP, SNAP, TANF



Submitting Verification: Missing verification documents was cited as one of the top two reasons for denials.

Benefits Claimant Intervention Moment



Administrator Intervention Moment

Reviewing: Determining if documents are legible and usesable.

Verify: Authenticating and and manually checking documents

Code: Extracting and inputting data

Use: Synthesizing data to calculate and compile wage reports.

Check out the full flow here

Summary of prototyping options

Pathway 1: Automated data extraction

Level of effort: High

Roadmap check: With a fully staffed team it will likely be mid 2026 before we are piloting.

Additional staffing needs: We will likely will need additional technical leadership (in addition to available staff) over next quarter.

Why start here: This pathway tackles the most complex and pressing problem first - automating data extraction would unlock broader document processing automation and significantly reduce administrative burden.

Opportunities:

- Wide applicability across multiple agencies and programs.
- Strong alignment with federal modernization goals: “Modernizing Federal Technology and Software to Maximize Efficiency and Productivity.”
- Compelling business case due to high-impact efficiency gains.

Risks:

- PII concerns related to temporary data storage.
- Higher technical complexity compared to submission-based solutions.
- Longer timeline before agencies see tangible improvements.

Pathway 2: Mobile-friendly document submission

Level of effort: Medium

Roadmap check: With a fully staffed team we could probably start piloting by end of this year if we found a partner.

Additional staffing needs: We could run a leaner technical team, but will likely need business dev/partnerships lead over next quarter.

Why start here: This pathway delivers faster impact, helping agencies that lack basic digital submission capabilities and ensuring document legibility without increasing processing burden.

Opportunities:

- Wide applicability across agencies.
- Faster realization of agency improvements with lower technical lift.
- Strong alignment with federal modernization goals: “Modernizing Federal Technology and Software to Maximize Efficiency and Productivity.”

Risks:

- PII concerns related to handling sensitive documents.
- Potential need for more system integrations to fully test in real-world settings.
- Weaker business case — while some agencies lack this capability, none have expressed an urgent need for a federal solution.

**We’ve talked to agencies who are missing this capability but none have felt quite so strongly about a federal solution.*

Potential Cost Savings for an automated data extraction tool

Processing could be 3x faster

Automation significantly reduces document processing times according to a RPA case study.

50M hours potentially saved

If we consider the total number of hours saved in the human services industry alone that's a total of 49,881,600 hours saved in a year*.

31% error rate found

Manual document flows lead to reduced in data accuracy (ABYY).

Development Roadmap

	Phase 1 Prototype design and technical discovery **Q2 2025**	Phase 2 Prototype testing and start technical build **Q3 2025**	Phase 3 Development **Q4 2025-Q1 2026**	Phase 4 Launch beta and find pilot partners **Q2 2026**
Technical Team	• Technical sprint team explores proof-of-concept for OCR tool • Feasibility go/no-go decision • Start LATO	• Outline technical roadmap • Build + test + iterate riskiest component • Align technical roadmap with prototype testing learnings • Continue steps for LATO	• Build prototype and iteratively improve • Develop testing environment • Create dummy documents for testing • Test prototype • Develop and measure quality testing metrics • Continue steps for LATO	• Onboard technical support team • Develop and test set-up instructions • Develop issue tracking process • Outline goals for development in beta • Resolve bugs and continuously improve product
Product and Design Team	• Onboard team of 5-6 technologists • Design front-end wireframes • Develop outreach and testing strategy • Sprint planning • Build relationships	• Reach out to state/local agency partners • Get feedback on wireframes from agency partners • Iterate on wireframes • Report findings to leadership for continued project support	• Design document submission website/portal • Test prototype with subset of agency partners • Develop outreach materials for pilot partnerships • Meet with Tech Law about PII concerns	• Develop demo video/deck/website • Develop a MOU template Support pilot agreement-process • Establish pilot engagement process • Define pilot phases and partnership targets • Continuously improve product frontend