Play 1.0	**Understand what people need**	
	Item	Method
	Early in the project, spend time with current and prospective users of the service	We spoke with both experienced Knowledge Management System users and novices to understand needs and desired functionality for the service.
	Use a range of qualitative and quantitative research methods to determine people’s goals, needs, and behaviors; be thoughtful about the time spent	We conducted surveys, issued questionnaires and conducted in-person and phone interviews to obtain feedback from the user community.
	Test prototypes of solutions with real people, in the field if possible	We asked users to provide feedback on clickable prototypes and wireframes where we incorporated their feedback into subsequent sprints and design evolutions. Usability testing was conducted early on in the process.
	Document the findings about user goals, needs, behaviors, and preferences	We created key experience pillar documents for both staff and admin users. These were shared with prospective users for feedback.
	Share findings with the team and agency leadership	Feedback was shared in weekly projects and following each round of interviews and questionnaires.
	Create a prioritized list of tasks the user is trying to accomplish, also known as “user stories”	Users stories are captured in the key experience pillar documents.
	As the digital service is being built, regularly test it with potential users to ensure it meets people’s needs	We have multiple touch points with prospectives users that ranged from conceptual design discussions to clickable prototype feedback.
Play 2.0	**Address the whole experience, from start to finish**	
	Item	Method
	Understand the different points at which people will interact with the service – both online and in person	The KMT was designed to accomodate roles for contributors, administators and public viewing personas.

	Identify pain points in the current way users interact with the service, and prioritize these according to user needs	We conducted interviews with KMT users and system providers to gather feedback on current usage patterns and best practices for KMT implementations.
	Design the digital parts of the service so that they are integrated with the offline touchpoints people use to interact with the service	The purpose of the service is to provide a repository for Knowledge Management Articles. These are stored in an online repository with approprate role based access.
	Develop metrics that will measure how well the service is meeting user needs at each step of the service	In our continuous user acceptance tests, we referenced and tracked key user experiences against, navigability, usability, time to accomplish simple tasks, content, gps reliability, and notification reliability and deliverability.
Play 3.0	**Make it simple and intuitive**	
	Item	Method
	Use a simple and flexible design style guide for the service. use the U.S. Web Design Standards as a default	A style guide was created to provide a framework for development and to ensure consistency across the application.
	Use the design style guide consistently for related digital services	A style guide was created to provide a framework for development and to ensure consistency across the application.
	Give users clear information about where they are in each step of the process	Our application is intuitive and straight forward with respect to creating, reading or administering Knowledge Articles.
	Follow accessibility best practices to ensure all people can use the service	We followed accessibility best practices based on Section 508 of the Americans with Disabilities Act and WCAG 2.0
	Provide users with a way to exit and return later to complete the process	Users can create a profile and come back and add or edit contact information,create KAs and review articles.
	Use language that is familiar to the user and easy to understand	We used simple and easy to follow language and logical user flow.
	Use language and design consistently throughout the service, including online and offline touch points	There were no offline touch points for this service, but the online service utilized a style guide was created to provide a framework for development and to ensure consistency across the application.
Play 4.0	**Build the service using agile and iterative practices**	
	Item	Method

	Ship a functioning “minimum viable product” (MVP) that solves a core user need as soon as possible, no longer than three months from the beginning of the project, using a “beta” or “test” period if needed	Our core focus from the beginning was to deliver a minimum viable product for the prototype. We focused on core functionality driven by user feedback for our initial release. We noted potential enhancements and functionality that could be added in future releases.
	Run usability tests frequently to see how well the service works and identify improvements that should be made	In addition to our automated test scripts, we performed manual smoke tests and functional unit tests to ensure usability and identify any design flaws.
	Ensure the individual's building the service communicate closely using techniques such as launch meetings, war rooms, daily standups, and team chat tools	The team conducted daily scrum meetings where key action items were discussed and potential roadblocks were identified. We relied on Aha! as our core ticketing system to manage our sprints and ensure proper burn down of tickets.We also used Slack as our internal communication tool to ensure everyone was apprised of progress and changes.
	Keep delivery teams small and focused; limit organizational layers that separate these teams from the business owners	We formed a small, agile team that met with the Business Sponsor and Product Manager on a daily basis
	Release features and improvements multiple times each month	We made several deployments over the course of the project using our continuous integration and continuous deployment methods setup using Docker and Jenkins. Deployments and automated test scripts were initiated each time a commit was made to a development branch.
	Create a prioritized list of features and bugs, also known as the “feature backlog” and “bug backlog”	Tickets were created for each feature set and requirements. A backlog of features and bugs were compiled and evaluated for inclusion at the beginning of each sprint to determine priority.
	Use a source code version control system	We utilized Github as our source control and code repository.
	Give the entire project team access to the issue tracker and version control system	All team members had access to Github (source control), Aha! (project management and ticket system) and Slack (team communication and collaboration).
	Use code reviews to ensure quality	Code reviews were initiating by the Technical Lead at predetermined points along the project to ensure consistency with architectural guidelines and coding standards.
Play 5.0	**Structure budgets and contract to support delivery**	

	Item	Method
	Budget includes research, discovery, and prototyping activities	We created an initial budget based on project scope and anticipated effort and tracked those through the burn down on our ticketing system.
	Contract is structured to request frequent deliverables, not multi-month milestones	We had a very focused project plan based on short sprints tied to specific project deliverables.
	Contract is structured to hold vendors accountable to deliverables	We delivered the project, Github and README.md file in accordance to the state's procurement requirements.
	Contract gives the government delivery team enough flexibility to adjust feature prioritization and delivery schedule as the project evolves	We built in user feedback loops into our delivery timeframe to account for changes in scope and feature evolution based on user feedback and preferences.
	Contract ensures open source solutions are evaluated when technology choices are made	We relied on several open source technologies to deliver the project.
	Contract specifies that software and data generated by third parties remains under our control, and can be reused and released to the public as appropriate and in accordance with the law	The entire project was deployed to a public Github repository with appropriate releases provided to end users.
	Contract allows us to use tools, services, and hosting from vendors with a variety of pricing models, including fixed fees and variable models like “pay-for-what-you-use” services	We have a number of pricing models that we employ including fixed fees based on project scope, annual license fees for unlimited usage or pay as you go in a Software as a Service (SaaS) fashion. Our preference is to align our interests with our clients.
	Contract specifies a warranty period where defects uncovered by the public are addressed by the vendor at no additional cost to the government	This project was limited to the delivery of a working prototype. There is no warranty period and the prototype is delivered "as is."
	Contract includes a transition of services period and transition-out plan	Please see above.

Play 6.0	Assign one leader and hold that person accountable	
	Item	Method
	A product owner has been identified	We identified Jason Connolly as the Product Owner and sole Business Sponsor. He retained full project authority over resources and technical decisions.
	All stakeholders agree that the product owner has the authority to assign tasks and make decisions about features and technical implementation details	All team members were in agreement that Mr. Connolly had final authority with respect to scope, features and technical implementation.
	The product owner has a product management background with technical experience to assess alternatives and weigh tradeoffs	Mr. Connolly has more than 18 years of experience in Product Management and has overseen the delivery of several enterprise applications with varying degrees of complexity, risk, size and scope. He is intimately familiar with the technical tradeoffs and alternative implementation methods sometimes needed to deliver a product to final completion.
	The product owner has a work plan that includes budget estimates and identifies funding sources	A resource pool and work estimates were established at the beginning of the project to track progress toward budget and timeline.
	The product owner has a strong relationship with the contracting officer	The team maintained an open communication line with the procurement department throughout the delivery of the prototype.

Play 7.0	Bring in experienced teams	
	Item	Method
	Member(s) of the team have experience building popular, high-traffic digital services	Our team has designed, built and managed systems that have administered more than $6 billion in federal aid and have been load tested to support several hundred thousand concurrent users.
	Member(s) of the team have experience designing mobile and web applications	Our team has designed, built and launched several mobile web, iOS and Android applications that are currently available for download and in use by major user bases.
	Member(s) of the team have experience using automated testing frameworks	Our team wrote automated tests scripts in Python. These were executed against the prototype as part of the Jenkins continuous integration and continuous deployment process. This was done in addition to our manual smoke and unit tests.

	Member(s) of the team have experience with modern development and operations (DevOps) techniques like continuous integration and continuous deployment	Our team utilized Docker containers and Jenkins to implement our continuous integration and continuous deployment process.
	Member(s) of the team have experience securing digital services	Our team has a long history designing, developing and maintaining highly secure environments that house critical and NPI data. We employ a series of tools including encryption, firewalls, intrusion detection devices, event monitoring services and alert logs to ensure the highest levels of compliance and security.
	A Federal contracting officer is on the internal team if a third party will be used for development work	N/A
	A Federal budget officer is on the internal team or is a partner	N/A
	The appropriate privacy, civil liberties, and/or legal advisor for the department or agency is a partner	Our internal legal counsel reviewed the prototype for conformance to privacy policies and intellectual property infringements and found no issues.
Play 8.0	**Choose a modern technology stack**	
	Item	Method
	Choose software frameworks that are commonly used by private-sector companies creating similar services	We utilized technologies and development languages that we routinely use for our commercial clients operating in the private sector.
	Whenever possible, ensure that software can be deployed on a variety of commodity hardware types	Our software was targeted for deployment on AWS. We created a Docker image for the backend and utilized a public image for MongoDB, both of which can be deployed to any hardware.
	Ensure that each project has clear, understandable instructions for setting up a local development environment, and that team members can be quickly added or removed from projects	Our Docker containers made the prototype easily portable to our different environments. In addition, source control through Github allows adding and removing of collaborators.
	Consider open source software solutions at every layer of the stack	We incorporated open source technologies at every level of our technology stack.

Play 9.0	Deploy in a flexible hosting environment	
	Item	Method
	Resources are provisioned on demand	The prototype is hosted through AWS which provides on demand provisioning of resources that can be scaled up or down based on need.
	Resources scale based on real-time user demand	This is accommodated through AWS.
	Resources are provisioned through an API	This is accommodated through AWS.
	Resources are available in multiple regions	This is accommodated through AWS.
	We only pay for resources we use	AWS only charges us for resources that are used.
	Static assets are served through a content delivery network	We utilized an S3 bucket to delivery static assets and content.
	Application is hosted on commodity hardware	The prototype is hosted through AWS.

Play 10.0	Automate testing and deployments	
	Item	Method
	Create automated tests that verify all user-facing functionality	Test scripts were written using Python to test back-end APIs and user-facing functions, respectively.
	Create unit and integration tests to verify modules and components	Unit and regression test scripts were executed against the prototype at various stages to ensure consistency and proper functionality.
	Run tests automatically as part of the build process	Automated test scripts were executed in conjunction with our continuous integration and continuous deployment process that was managed by Jenkins.
	Perform deployments automatically with deployment scripts, continuous delivery services, or similar techniques	Builds were automatically created and deployed using Jenkins. This process was kicked off each time a commitment was made to a development branch.
	Conduct load and performance tests at regular intervals, including before public launch	Load tests were out of scope for the delivery of the prototype. However, we have utilized Jmeter for load testing in prior projects.

Play 11.0	**Manage security and privacy through reusable processes**	
	Item	Method
	Contact the appropriate privacy or legal officer of the department or agency to determine whether a System of Records Notice (SORN), Privacy Impact Assessment, or other review should be conducted	We reviewed the project with our internal legal team and determined that there were no specific legal or privacy requirements required for the prototype.
	Determine, in consultation with a records officer, what data is collected and why, how it is used or shared, how it is stored and secured, and how long it is kept	For the purposes of the prototype, we are collecting Knowledge Article information.
	Determine, in consultation with a privacy specialist, whether and how users are notified about how personal information is collected and used, including whether a privacy policy is needed and where it should appear, and how users will be notified in the event of a security breach	We reviewed the project with our internal legal team and determined that there were no specific legal or privacy requirements required for the prototype.
	Consider whether the user should be able to access, delete, or remove their information from the service	Functionality was provided to edit or remove user input information or to unsubscribe from the service.
	"Pre-certify" the hosting infrastructure used for the project using FedRAMP	AWS is a pre-certified IaaS provider on FedRAMP.
	Use deployment scripts to ensure configuration of production environment remains consistent and controllable	We used Docker containers and Jenkins for continuous integration and continuous deployment to ensure consistency among deployments to our various environments.
Play 12.0	**Use data to drive decisions**	

	Item	Method
	Monitor system-level resource utilization in real time	We relied on AWS Cloudwatch to monitor system availability and performance.
	Monitor system performance in real-time (e.g. response time, latency, throughput, and error rates)	We relied on AWS Cloudwatch to monitor system availability and performance.
	Ensure monitoring can measure median, 95th percentile, and 98th percentile performance	AWS Cloudwatch provides detailed metrics on performance to evaluate uptime percentiles.
	Create automated alerts based on this monitoring	AWS Cloudwatch notifications were setup to identify degradation in services and when certain system parameter thresholds were triggered.
	Track concurrent users in real-time, and monitor user behaviors in the aggregate to determine how well the service meets user needs	This was out of scope for the prototype, but normally we rely on analytics derived using Firebase, Google Analytics or Crashlytics.
	Publish metrics internally	These were published and accessible via the AWS Cloudwatch dashboard.
	Publish metrics externally	N/A
	Use an experimentation tool that supports multivariate testing in production	N/A
Play 13.0	**Bring in experienced teams**	
	Item	Method
	Offer users a mechanism to report bugs and issues, and be responsive to these reports	During regular projects we set up an Idea Portal within our Aha! ticketing system that provides clients with the ability to create new feature and bug tickets that go directly to the project teams. These tickets can be voted upon and flushed out in more detail and then pulled into subsequent sprints and deployments. However, this was not setup for the prototype given the project timeframe and limited interaction with users post launch.

	Provide datasets to the public, in their entirety, through bulk downloads and APIs (application programming interfaces)	Knowledge Arcticles tagged as Public are available for viewing on the platform without the necessity of a login.
	Ensure that data from the service is explicitly in the public domain, and that rights are waived globally via an international public domain dedication, such as the "Creative Commons Zero" waiver	All information included in the prototype was publicly available.
	Catalog data in the agency's enterprise data inventory and add any public datasets to the agency's public data listing	This is out of scope for the prototype (N/A).
	Ensure that we maintain the rights to all data developed by third parties in a manner that is releasable and reusable at no cost to the public	All information included in the prototype was publicly available.
	Ensure that we maintain contractual rights to all custom software developed by third parties in a manner that is publishable and reusable at no cost	All source code was deployed to a public Github repository.
	When appropriate, create an API for third parties and internal users to interact with the service directly	This is out of scope for the prototype (N/A).
	When appropriate, publish source code of projects or components online	All source code was deployed to a public Github repository.
	When appropriate, share your development process and progress publicly	Our entire process and documentation was fully documented in the README.md file contained in our Github repository.