

Play 1: Understand What People Need	
Checklist	What we did
Early in the project, spend time with current and prospective users of the service	We defined scenarios that modeled a normal day in the life of the end user, we performed an interview and held a focus group session with our users
Use a range of qualitative and quantitative research methods to determine people's goals, needs, and behaviors; be thoughtful about the time spent	As a result of the focus group and interview, we were able to create personas that represented the user profiles and high-level user stories(scenarios). These stories captured the basic requirements for the application which allowed the team to brainstorm and collaboratively create storyboards. From the storyboard session, we took the results and prototyped them into wireframes and incorporated the feedback from the wireframe walkthrough to create the Agile user stories that represented the set of application requirements. At the end of each sprint, we conducted a sprint review where we would demo the application to the product owner and users and allow them to work with the application and provide feedback. The iterative feedback helped us to validate that the application's usability and behavior was inline with the user's goals and expectations
Test prototypes of solutions with real people, in the field if possible	We conducted Usability testing with our users to gather their feedback. Additionally, at the end of each sprint, we reviewed the application at the sprint review meeting
Document the findings about user goals, needs, behaviors, and preferences	We elicited feedback while conducting focus groups. User stories were updated using this feedback
Share findings with the team and agency leadership	Findings were discussed at the Sprint Reviews and our bi-weekly team collaboration meeting
Create a prioritized list of tasks the user is trying to accomplish, also known as "user stories"	Our user stories and our backlog were managed in JIRA. The backlog schedule assignment and status were updated during every sprint.
As the digital service is being built, regularly test it with potential users to ensure it meets people's needs	We invited users to all of our sprint demos and they provided valuable feedback at every stage of the project, from ideation to wireframes to coded prototype. They also performed usability testing of our AWS deployed prototype
Key Questions	Our Response

Who are your primary users?	Employees of State Agencies/Departments who would capture, manage, and share knowledge articles - that would make it quick and easy for other State Agencies/Department users to search and view the articles
What user needs will this service address?	This service will help in capturing organizational knowledge (including documents, user-configurable forms, tables, and workflows) and effectively managing and sharing it across the State Agencies/Departments
Why does the user want or need this service?	It will help to capture existing knowledge assets from diversified sources such as existing documents, procedures, and people that can be managed and shared across the state departments; this would also enhance the State's ability to protect its key knowledge and competencies from being lost or copied
Which people will have the most difficulty with the service?	We have not yet addressed the problem of people who don't use smart phones or have no experience at all with computers and internet
Which research methods were used?	The following user research methods were used: User Interviews, Focus Group, Personas, Empathy Map, Storyboards, Wireframes.
What were the key findings?	Key findings included the requirement for an easily navigable interface which could be accessible through desktop or mobile devices. The findings were documented and shared with the team members. We determined that the Prototype requires three basic functions/epics: create article, approve article and search/view article
How were the findings documented? Where can future team members access the documentation?	All of our work and findings are publicly documented on GitHub
How often are you testing with real people?	We tested with real people during each of our sprints
Play 2: Address the Whole Experience, from Start to Finish	
Checklist	What we did

Understand the different points at which people will interact with the service – both online and in person	We interviewed several users to get a clearer picture of their needs and how they would prefer to interact with the system. Based on the user interviews, we created end-to-end process flow for the lifecycle of Knowledge Articles and identified the user roles, activities and corresponding touchpoints with the business process. The process also identifies the notification requirements on the event of the change in the Knowledge Article lifecycle state. From the user interview, it was quite obvious that the service needs to be delivered through both Web and Mobile channels. We built a completely responsive site and would propose building a separate mobile app in the future so users can get immediate notifications about new messages.
Identify pain points in the current way users interact with the service, and prioritize these according to user needs	Users do not have much experience using a similar portal as it does not exist currently. The user pain points were captured during the User Interview process and the user stories were refined and prioritized to address the user needs
Design the digital parts of the service so that they are integrated with the offline touch points people use to interact with the service	N/A to the Working Prototype;
Develop metrics that will measure how well the service is meeting user needs at each step of the service	We measured usability of navigation, comprehension of content and usability
Key Questions	Our Response
What are the different ways (both online and offline) that people currently accomplish the task the digital service is designed to help with?	Currently there is no centralized repository to capture and manage organizational knowledge. Employees need to access existing knowledge from scattered physical and electronic documents and other employees who are knowledgeable in the specific subject area

Where are user pain points in the current way people accomplish the task?	There is no centralized repository to capture and manage organizational knowledge; Difficult to get information on a timely manner; Need to wait for the availability of other employees for information about specific subject matter thus resulting in loss of time, resources and human effort; Do not have a system to document, store and manage the technology and architecture knowledge base that can be shared with various users within state departments; No process to validate the authenticity, accuracy and completeness of the information being shared
Where does this specific project fit into the larger way people currently obtain the service being offered?	The working prototype can be enhanced further and deployed and managed by CDT that can be used by CDT employees and other departments (customers) of the State of California
What metrics will best indicate how well the service is working for its users?	Metrics that would indicate the utility of the service would be: number of active users on the knowledge portal, percentage of CDT employees visiting the knowledge portal, number of other state department employees actively using the knowledge portal, number of returning users, number of views and rating of the articles, total number of articles in the knowledge repository, trend in new article addition to the repository, etc.
Play 3: Make it Simple and Intuitive	
Checklist	What we did
Use a simple and flexible design style guide for the service	We implemented a simple and responsive Design Style Guide adapted from US Web Design Standards, as well as adhering to goals and principles of Material Design
Use the design style guide consistently for related digital services	The style guide was used consistently when designing application user interfaces
Give users clear information about where they are in each step of the process	Our application displays clear and concise headings and sub-headings and navigation for users to understand what is to be done next. In addition, the application displays different inbox for the article queue that are in draft, in review state to effectively guide different user roles about their tasks.
Follow accessibility best practices to ensure all people can use the service	We have followed WCAG 2.0 compliance guidelines and tested for Section 508 compliance and WCAG using SaaS tool, Cinthiasays.com.

Provide users with a way to exit and return later to complete the process	The working prototype uses various mechanisms to help the users to exit and comeback at a later stage to complete the tasks. Few of those mechanisms include ability for article authors to save the new and incomplete articles in draft state that can be completed and submitted for approval at a later stage; the application provides queues for the draft and in review articles to the authors and approvers so they can quickly view the queued articles and take necessary actions; the application send automated email notification to the appropriate users on the event of change in the article state so that the users can login to the application at a later stage and take necessary actions
Use language that is familiar to the user and easy to understand	We have used simple language in the prototype which is understood by people
Use language and design consistently throughout the service, including online and offline touch points	We have used language and design consistently throughout the service enforced through the Style Guide
Key Questions	Our Response
What primary tasks are the user trying to accomplish?	Establish a web-based application that allows users to a) login to the application b) create knowledge article c) update article d) review article e) publish article f) search and read article
Is the language as plain and universal as possible?	We have performed multiple reviews with the end users to validate that the language used in the prototype is simple and easy to understand
What languages is your service offered in?	Our service is offered in English
If a user needs help while using the service, how do they go about getting it?	We have included a User Manual for the user to be guided through the application
How does the service's design visually relate to other government services?	We have included CA.gov and California Department of Technology seal in the home page header
Play 4: Build the Service Using Agile and Iterative Practices	
Checklist	What we did
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	We had our MVP implemented in 4 weeks time which includes key user stories and other requirements specified in the RFI.

Run usability tests frequently to see how well the service works and identify improvements that should be made	We carried out usability test every few days and identified the improvement areas and then implemented those improvements
Ensure the individuals building the service communicate closely using techniques such as launch meetings, war rooms, daily standups, and team chat tools	We conducted daily meetings in person where we communicated and collaborated within the team to understand and incorporate user feedback; in addition we used team collaboration tools such as JIRA, GitHub and Trello
Keep delivery teams small and focused; limit organizational layers that separate these teams from the business owners	Our team comprised one product manager, one technical architect, three developers, one content designer, one delivery manager, one usability tester and one agile coach
Release features and improvements multiple times each month	We conducted a total of 5 sprints over the total time period to understand areas of improvements
Create a prioritized list of features and bugs, also known as the “feature backlog” and “bug backlog”	We had created feature backlogs for user stories bug backlog list for defects in JIRA
Use a source code version control system	We have used Github as our source code repository and used DockerHub as Docker image repository
Give the entire project team access to the issue tracker and version control system	The entire team has access to the issue tracker(JIRA) and version control system(GitHub)
Use code reviews to ensure quality	Code reviews were conducted amongst our developer team
Key Questions	Our Response
How long did it take to ship the MVP? If it hasn't shipped yet, when will it?	It took us 4 weeks to ship the MVP
How long does it take for a production deployment?	For the first time deployment in Production environment, it takes 2-3 hours for initial configuration of the CI/CD framework and testing; after the CI/CD is configured, the deployment in production environment is automated and deployment happens at the configured schedule
How many days or weeks are in each iteration/sprint?	Five days per sprint
Which version control system is being used?	GitHub

How are bugs tracked and tickets issued? What tool is used?	We create issue tasks in JIRA to track defects
How is the feature backlog managed? What tool is used?	We had created feature backlogs for user stories bug backlog list for defects in JIRA
How often do you review and reprioritize the feature and bug backlog?	Once before each sprint starts as per of Sprint Planning
How do you collect user feedback during development? How is that feedback used to improve the service?	In addition to scheduling reviews with subject matter experts that approximate users, the prototype was made continuously available to the entire team who were instructed to utilize all features of the system to evaluate quality and completeness based on the MVP. The team then provided suggestions for improvement in the form of new user stories that were either incorporated into the backlog for the following sprint or for future consideration.
At each stage of usability testing, which gaps were identified in addressing user needs?	Two gaps were identified. 1) Self service username and password reset 2) Self service user registration. The Product Manager determined that these two gaps could be developed as future enhancements to the product since they are not necessary for MVP.
Play 5: Structure Budgets and Contracts to Support Delivery	
Checklist	What we did
Budget includes research, discovery, and prototyping activities	We had initially created an internal budget for this project and managed the budget over the entire project lifecycle
Contract is structured to request frequent deliverables, not multi-month milestones	N/A
Contract is structured to hold vendors accountable to deliverables	N/A
Contract gives the government delivery team enough flexibility to adjust feature prioritization and delivery schedule as the project evolves	N/A
Contract ensures open source solutions are evaluated when technology choices are made	N/A

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	N/A
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	N/A
Contract specifies a warranty period where defects uncovered by the public are addressed by the vendor at no additional cost to the government	N/A
Contract includes a transition of services period and transition-out plan	N/A
Key Questions	Our Response
What is the scope of the project? What are the key deliverables?	The scope of this working prototype is implement a Knowledge Management Tool to capture, manage and share Knowledge Articles following US Digital Services Playboog guidelines.
What are the milestones? How frequent are they?	For this working prototype, we have milestones defined for each of the five sprints in JIRA as Sprint goal.
What are the performance metrics defined in the contract (e.g., response time, system uptime, time period to address priority issues)?	Not applicable for this prototype.
Play 6: Assign One Leader and Hold that Person Accountable	
Checklist	What we did
A product owner has been identified	Sanjib Nayak has been designated as the Product Owner
All stakeholders agree that the product owner has the authority to assign tasks and make decisions about features and technical implementation details	Our team members have worked with Sanjib earlier and they understood that the product owner had the authority to assign tasks and make decisions about features and technical implementation details

The product owner has a product management background with technical experience to assess alternatives and weigh tradeoffs	Sanjib brings 20 years of experience in Product Development, Product Management and Technology Solutions.
The product owner has a work plan that includes budget estimates and identifies funding sources	Product owner had created a budgeting plan in the first sprint and had a detailed work plan to which the team adhered
The product owner has a strong relationship with the contracting officer	N/A
Key Questions	Our Response
Who is the product owner?	Sanjib Nayak
What organizational changes have been made to ensure the product owner has sufficient authority over and support for the project?	No organizational change was required in order to give the Product Owner sufficient authority and to provide support for the project
What does it take for the product owner to add or remove a feature from the service?	The product owner had authority to add/delete/modify features based on user feedback
Play 7: Bring in Experienced Teams	
Checklist	What we did
Member(s) of the team have experience building popular, high-traffic digital services	Our scrum team brings over 100 years of combined experience in designing, and implementing high-traffic digital services using modern solution architecture, open source technologies and open standards
Member(s) of the team have experience designing mobile and web applications	Our technical team members have extensive experience in designing and developing extensive web applications that can be delivered through diversified devices and browsers
Member(s) of the team have experience using automated testing frameworks	Our technical team has extensive experience using modern automated, unit, and functional testing tools such as Jasmine, Karma, Selenium etc.
Member(s) of the team have experience with modern development and operations (DevOps) techniques like continuous integration and continuous deployment	Our technical team has extensive experience in using modern DevOps techniques such as infrastructure automation and CI/CD using technologies such as Docker, Kubernetes, DockerHub and Jenkins etc.

Member(s) of the team have experience securing digital services	Our team has extensive experience in risk-based security and privacy control(NIST), security classification of system and data, designing and implementing security controls based on security classification of the system. For the working prototype, we have implemented username/password based authentication, role-based access control and API security using security tokens
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	N/A
Play 8: Choose a Modern Technology Stack	
Checklist	What we did
Choose software frameworks that are commonly used by private-sector companies creating similar services	We implemented the working prototype solution using modern Microservice based Service Oriented Architecture (SOA) that is implemented using commonly used Open source technologies and Open standards as described in our Technical Approach section
Whenever possible, ensure that software can be deployed on a variety of commodity hardware types	The entire solution is containerized using Docker containers that enables the application to be platform agnostic
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	We have established automated deployment of the platform in various SDLC environments using Docker, DockerHub and Jenkins. This enables developer to establish their local environment quickly
Consider open source software solutions at every layer of the stack	We implemented the working prototype solution using modern Microservice based Service Oriented Architecture (SOA) that is implemented using commonly used Open source technologies and Open standards as described in our Technical Approach section
Key Questions	Our Response
What is your development stack and why did you choose it?	We chose Open source and Open standard based technology stack that provides necessary technical capabilities for the implementation of the Knowledge Management Portal solution. The solution architecture and the technology stack used for the implementation of the working prototype are described in the Technical Approach section

Which databases are you using and why did you choose them?	We are using Open Source relational database, PostgreSQL to store the relational data and file system(AWS S3 in production) to store uploaded knowledge articles
How long does it take for a new team member to start developing?	We selected team members from within our existing technical team who are already familiar with the technology stacks and had worked together in the past; so each developer on the team was able to start developing from day one.
Play 9: Deploy in a Flexible Hosting Environment	
Checklist	What we did
Resources are provisioned on demand	For the Working Prototype, we used the Amazon Web Services (AWS EC2, AWS S3, AWS SES) to provide hosting, platform and infrastructure support. AWS provides resources that can be provisioned dynamically, as user demand changes for production applications. We can allocate resources either via a web management console or via API
Resources scale based on real-time user demand	AWS supports this option
Resources are provisioned through an API	AWS supports this option
Resources are available in multiple regions	AWS supports this option
We only pay for resources we use	Yes
Static assets are served through a content delivery network	N/A
Application is hosted on commodity hardware	Yes
Key Questions	Our Response
Where is your service hosted?	For the Working Prototype, we used the Amazon Web Services (AWS EC2, AWS S3, AWS SES) to provide hosting, platform and infrastructure support. AWS provides resources that can be provisioned dynamically, as user demand changes for production applications. We can allocate resources either via a web management console or via API
What hardware does your service use to run?	AWS EC2
What is the demand or usage pattern for your service?	N/A

What happens to your service when it experiences a surge in traffic or in load?	N/A
How much capacity is available in your hosting environment?	Amazon's cloud services are saleable based on capacity. This does not apply to the Working Prototype.
How long does it take you to provision a new resource, like an application server?	Approximately five (5) minutes to provision a new application server.
How have you designed your service to scale based on demand?	The N-tier distributed architecture supports both horizontal and vertical scalability
How are you paying for your hosting infrastructure (e.g., by the minute, hourly, daily, monthly, fixed)?	By the minute although AWS offers a fixed cost if reserving for a minimum of one year.
Is your service hosted in multiple regions, availability zones, or data centers?	N/A
In the event of a catastrophic disaster to a datacenter, how long will it take to have the service operational?	N/A
What would be the impact of a prolonged downtime window?	N/A
What data redundancy do you have built into the system, and what would be the impact of a catastrophic data loss?	N/A
How often do you need to contact a person from your hosting provider to get resources or to fix an issue?	N/A
Play 10: Automate Testing and Deployments	
Checklist	What we did
Create automated tests that verify all user-facing functionality	Automated functional testing was conducted using Selenium
Create unit and integration tests to verify modules and components	Automated unit testing was conducted using Jasmine and Karma and automated integration testing was conducted using Selenium and automated API testing was performed using CURL script

Run tests automatically as part of the build process	The automated Unit testing script and automated API testing was performed using CURL script as part of the build process
Perform deployments automatically with deployment scripts, continuous delivery services, or similar techniques	We implemented modern DevOps techniques such as infrastructure automation and CI/CD using technologies such as Docker, DockerHub and Jenkins
Conduct load and performance tests at regular intervals, including before public launch	Automated load testing was not performed as part of the Working Prototype scope and added as backlog for future iterations
Key Questions	Our Response
How long does it take to build, test, and deploy a typical bug fix?	We have automated the entire build, test and deployment pipeline. As soon as the code for the new feature checked-in into GitHub, the build, test and deploy tasks kicks-off instantly. Assuming that the test scripts are created for the new features and the testing is passed as a pipeline task the entire process can be completed automatically in few minutes.
How long does it take to build, test, and deploy a new feature into production?	We have automated the entire build, test and deployment pipeline. As soon as the code for the new feature checked-in into GitHub, the build, test and deploy tasks kicks-off instantly. Assuming that the test scripts are created for the new features and the testing is passed as a pipeline task the entire process can be completed automatically in few minutes.
How frequently are builds created?	Scheduled automated deployment once a day in the night
What test tools are used?	Unit Testing - Jasmine, Karma Automated Functional Testing - Selenium Smoke Testing - CURL
Which deployment automation or continuous integration tools are used?	Deployment Automation - Docker, DockerHub, Jenkins Continuous Integration - Jenkins
What is the estimated maximum number of concurrent users who will want to use the system?	N/A
How many simultaneous users could the system handle, according to the most recent capacity test?	N/A
How does the service perform when you exceed the expected target usage volume? Does it degrade gracefully or catastrophically?	N/A

What is your scaling strategy when demand increases suddenly?	The application is deployed on AWS and the infrastructure can scale when demand increases.
Play 11: Manage Security and Privacy through Reusable Processes	
Checklist	What we did
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	Our Security architect assessed the Security Classification of the Data and the system using NIST guidelines and recommended necessary controls for the working prototype. The system doesn't capture, store and share any PI, PII PHI and FTI data. The security Classification for the KMP is LOW.
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	We consulted with the our Security Engineer who acted as Records Officer for this Prototype.
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	No personal information is collected and shared by the Knowledge Management Portal
Consider whether the user should be able to access, delete, or remove their information from the service	N/A. No personal information is collected and shared by the Knowledge Management Portal
"Pre-certify" the hosting infrastructure used for the project using FedRAMP	N/A - Not a requirement for this RFI.
Use deployment scripts to ensure configuration of production environment remains consistent and controllable	We are using aotomated CI/CD scripts implemented using Jenkins that is used for automated build and deployment of code in various SDLC environments.
Key Questions	Our Response

Does the service collect personal information from the user? How is the user notified of this collection?	NO
Does it collect more information than necessary? Could the data be used in ways an average user wouldn't expect?	NO
How does a user access, correct, delete, or remove personal information?	N/A
Will any of the personal information stored in the system be shared with other services, people, or partners?	N/A
How and how often is the service tested for security vulnerabilities?	This question does not apply to the Working Prototype.
How can someone from the public report a security issue?	This question does not apply to the Working Prototype.
Play 12: Use Data to Drive Decisions	
Key Questions	Our Response
What are the key metrics for the service?	This question does not apply to the Working Prototype.
How have these metrics performed over the life of the service?	This question does not apply to the Working Prototype.
Which system monitoring tools are in place?	We are using AWS Cloud Watch for continuous monitoring.
What is the targeted average response time for your service? What percent of requests take more than 1 second, 2 seconds, 4 seconds, and 8 seconds?	Our target average response time for all the transactions is less than 2 sec.
What is the average response time and percentile breakdown (percent of requests taking more than 1s, 2s, 4s, and 8s) for the top 10 transactions?	This question does not apply to the Working Prototype. However we observed average response time for all the transactions about 90ms during our testing.

What is the volume of each of your service's top 10 transactions? What is the percentage of transactions started vs. completed?	This question does not apply to the Working Prototype.
What is your service's monthly uptime target?	This question does not apply to the Working Prototype.
What is your service's monthly uptime percentage, including scheduled maintenance? Excluding scheduled maintenance?	This question does not apply to the Working Prototype.
How does your team receive automated alerts when incidents occur?	We are using AWS Cloud Watch for continuous monitoring. The AWS Cloud Watch tool send email notification whenever there is an issue (process, momery, disk space etc.).
How does your team respond to incidents? What is your post-mortem process?	We are using AWS Cloud Watch for continuous monitoring. The AWS Cloud Watch tool send email notification whenever there is an issue (process, momery, disk space etc.). When our team receives incident notifications, they perform root cause analysis and take necessary action to address the issue.
Which tools are in place to measure user behavior?	This question does not apply to the Working Prototype.
What tools or technologies are used for A/B testing?	A/B testing has not been performed for the prototype.
How do you measure customer satisfaction?	This question does not apply to the Working Prototype.
Play 13: Default to open	
Key Questions	Our Response
How are you collecting user feedback for bugs and issues?	JIRA was the tool we used to report bugs and issues.
If there is an API, what capabilities does it provide? Who uses it? How is it documented?	Our application uses API First approach and data is accessed through the RESTful API interfaces. The APIs are accessed through an API Gateway by the Business application serving the web and mobile channels.
If the codebase has not been released under an open source license, explain why.	N/A. The codebase has been released to the GitHub directory as a public project.
What components are made available to the public as open source?	The entire project is available to the public as open source.

What datasets are made available to the public?	The Knowledge article data is managed and shared to the registered Department users and the Role Based Access Control is used to control accessibility of various functions through the application.
---	--