

Evidence that EZRxLookup follows the U.S. Digital Services Playbook

Contents

Play 1: Understand what people need.....	2
Play 2: Address the whole experience, from start to finish.....	3
Play 3: Make it simple and intuitive.....	4
Play 4: Build the service using agile and iterative practices	5
Play 5: Structure budgets and contracts to support delivery	6
Play 6: Assign one leader and hold that person accountable	7
Play 7: Bring in experienced teams	8
Play 8: Choose a modern technology stack	9
Play 9: Deploy in a flexible hosting environment.....	10
Play 10: Automate testing and deployments.....	11
Play 11: Manage security and privacy through reusable processes	12
Play 12: Use data to drive decisions.....	14
Play 13: Default to open	15

Play 1: Understand what people need	
Checklist	
1. Early in the project, spend time with current and prospective users of the service	x
2. Use a range of qualitative and quantitative research methods to determine people's goals, needs, and behaviors; be thoughtful about the time spent	x
3. Test prototypes of solutions with real people, in the field if possible	x
4. Document the findings about user goals, needs, behaviors, and preferences	x
5. Share findings with the team and agency leadership	x
6. Create a prioritized list of tasks the user is trying to accomplish, also known as "user stories"	x
7. As the digital service is being built, regularly test it with potential users to ensure it meets people's needs	x
Key Questions	Evidence
1. Who are your primary users?	General public. Personas were developed for a Senior Citizen, Parent with child, and Single Self-Diagnoser.
2. What user needs will this service address?	Quick status on whether a drug has been recalled and/or if adverse effects have been reported.
3. Why does the user want or need this service?	Quick feedback on a drug.
4. Which people will have the most difficulty with the service?	Someone who may want statistical data or more details on derogatory drug data.
5. Which research methods were used?	Interviews, reviewing other similar websites, analysis of openFDA data and API, and discovery sessions.

6. What were the key findings?	Other tools/websites can be overwhelming and do not provide simple, succinct results. Users need a fast, simple, and reliable tool to determine whether to purchase or consume a drug.
7. How were the findings documented? Where can future team members access the documentation?	Findings were documented in the Personas, Journey map, mission statement and strategy described in the System Flow Document, and the Product Backlog.
8. How often are you testing with real people?	Daily.
Play 2: Address the whole experience, from start to finish	
Checklist	
1. Understand the different points at which people will interact with the service – both online and in person	x
2. Identify pain points in the current way users interact with the service, and prioritize these according to user needs	x
3. Design the digital parts of the service so that they are integrated with the offline touch points people use to interact with the service	x
4. Develop metrics that will measure how well the service is meeting user needs at each step of the service	x
Key Questions	Evidence
1. What are the different ways (both online and offline) that people currently accomplish the task the digital service is designed to help with?	Users are currently trying to dig through multiple websites and information available on the internet.
2. Where are user pain points in the current way people accomplish the task?	The current pain point is that there is an overwhelming amount of data to sort through in order to make a reasonable conclusion about the safety of a drug based on available information.

3. Where does this specific project fit into the larger way people currently obtain the service being offered?	This project will provide a fast, simple, and reliable tool for users to leverage when determining whether to consume or purchase a drug.
4. What metrics will best indicate how well the service is working for its users?	Website hits with unique IPs/return visits, response time for suggestions, response time of search, and successful hits per session.

Play 3: Make it simple and intuitive

Checklist	
1. Create or use an existing, simple, and flexible design style guide for the service	x
2. Use the design style guide consistently for related digital services	x
3. Give users clear information about where they are in each step of the process	x
4. Follow accessibility best practices to ensure all people can use the service	x
5. Provide users with a way to exit and return later to complete the process	N/A
6. Use language that is familiar to the user and easy to understand	x
7. Use language and design consistently throughout the service, including online and offline touch points	x
Key Questions	
1. What primary tasks are the user trying to accomplish?	Search for a drug, determine the status of a drug, and view a summary of adverse events.
2. Is the language as plain and universal as possible?	Yes, and visual indicators (e.g. a green check mark or a red "X") are used to further simplify the process so the user does not have to read any derogatory drug details if they chose not to.
3. What languages is your service offered in?	English

4. If a user needs help while using the service, how do they go about getting it?	This is a prototype, help functions were deferred and are in the Product Backlog.
5. How does the service's design visually relate to other government services?	Other Government websites were reviewed and the <i>Mozilla Style Guide</i> and the <i>Franklin Covey Style Guide: For Business and Technical Communication</i> were followed.

Play 4: Build the service using agile and iterative practices

Checklist	
1. 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	x
2. Run usability tests frequently to see how well the service works and identify improvements that should be made	x
3. Ensure the individuals building the service communicate closely using techniques such as launch meetings, war rooms, daily standups, and team chat tools	x
4. Keep delivery teams small and focused; limit organizational layers that separate these teams from the business owners	x
5. Release features and improvements multiple times each month	x
6. Create a prioritized list of features and bugs, also known as the "feature backlog" and "bug backlog"	x
7. Use a source code version control system	x
8. Give the entire project team access to the issue tracker and version control system	x
9. Use code reviews to ensure quality	x
Key Questions	Evidence

1. How long did it take to ship the MVP? If it hasn't shipped yet, when will it?	Six (6) days.
2. How long does it take for a production deployment?	Approximately two (2) minutes.
3. How many days or weeks are in each iteration/sprint?	An iteration is eight (8) hours.
4. Which version control system is being used?	GitHub.
5. How are bugs tracked and tickets issued? What tool is used?	Bugs and tickets are issued and tracked in LeanKit.
6. How is the feature backlog managed? What tool is used?	The Feature backlog is managed with LeanKit.
7. How often do you review and reprioritize the feature and bug backlog?	Daily.
8. How do you collect user feedback during development? How is that feedback used to improve the service?	User Feedback is collected daily after the Review and Stakeholder demos. This feedback is then analyzed and incorporated into Features, User Stories, Acceptance Criteria, and/or tasks in LeanKit via Usability Test Reports.
9. At each stage of usability testing, which gaps were identified in addressing user needs?	Usability Test Reports helped us identify and address user needs, including simple search, drug status and mobile view.

Play 5: Structure budgets and contracts to support delivery

Checklist	
1. Budget includes research, discovery, and prototyping activities	x
2. Contract is structured to request frequent deliverables, not multi-month milestones	N/A
3. Contract is structured to hold vendors accountable to deliverables	N/A
4. Contract gives the government delivery team enough flexibility to adjust feature prioritization and delivery schedule as the project evolves	N/A

5. Contract ensures open source solutions are evaluated when technology choices are made	N/A
6. 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
7. 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
8. 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
9. Contract includes a transition of services period and transition-out plan	N/A
Key Questions	Evidence
1. What is the scope of the project? What are the key deliverables?	The scope of the project is to produce a prototype leveraging openFDA data with open source technologies and agile development methodologies. The key deliverables include a working prototype, repository containing code and artifacts.
2. What are the milestones? How frequent are they?	A milestone is the acceptance of user stories in a given iteration. Each iteration is eight (8) hours.
3. What are the performance metrics defined in the contract (e.g., response time, system uptime, time period to address priority issues)?	N/A; no contract at this time.
Play 6: Assign one leader and hold that person accountable	
Checklist	
1. A product owner has been identified	x

2. All stakeholders agree that the product owner has the authority to assign tasks and make decisions about features and technical implementation details	x
3. The product owner has a product management background with technical experience to assess alternatives and weigh tradeoffs	x
4. The product owner has a work plan that includes budget estimates and identifies funding sources	x
5. The product owner has a strong relationship with the contracting officer	x
Key Questions	Evidence
1. Who is the product owner?	The Product Owner is Kion Yap (B&A's CEO), who assigned John Poling as the Product Manager.
2. What organizational changes have been made to ensure the product owner has sufficient authority over and support for the project?	No organizational changes were necessary as Kion Yap (Product Owner) has authority for B&A.
3. What does it take for the product owner to add or remove a feature from the service?	The Product Owner works with the users and stakeholders during daily planning sessions to prioritize iteration objectives.
Play 7: Bring in experienced teams	
Checklist	
1. Member(s) of the team have experience building popular, high-traffic digital services	x
2. Member(s) of the team have experience designing mobile and web applications	x
3. Member(s) of the team have experience using automated testing frameworks	x
4. Member(s) of the team have experience with modern development and operations (DevOps) techniques like continuous integration and continuous deployment	x

5. Member(s) of the team have experience securing digital services	x
6. A Federal contracting officer is on the internal team if a third party will be used for development work	N/A
7. A Federal budget officer is on the internal team or is a partner	N/A
8. 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	
1. Choose software frameworks that are commonly used by private-sector companies creating similar services	x
2. Whenever possible, ensure that software can be deployed on a variety of commodity hardware types	x
3. 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	x
4. Consider open source software solutions at every layer of the stack	x
Key Questions	Evidence
1. What is your development stack and why did you choose it?	<p>Java, Spring, and Elasticsearch. Java and Spring were chosen because they are open source, commonly used, powerful, and flexible.</p> <p>Elasticsearch was chosen because it is open source, it is already being used by openFDA, and it is quick and easy to implement search criteria suggestions.</p>

2. Which databases are you using and why did you choose them?	Elasticsearch was chosen because it is open source, it is already being used by openFDA, and it is quick and easy to implement search criteria suggestions. All other data access is consumed via openFDA provided APIs.
3. How long does it take for a new team member to start developing?	Right away, after the Development environment is established.

Play 9: Deploy in a flexible hosting environment

Checklist	
1. Resources are provisioned on demand	x
2. Resources scale based on real-time user demand	x
3. Resources are provisioned through an API	x
4. Resources are available in multiple regions	x
5. We only pay for resources we use	x
6. Static assets are served through a content delivery network	x
7. Application is hosted on commodity hardware	x
Key Questions	Evidence
1. Where is your service hosted?	Amazon Web Services (AWS).
2. What hardware does your service use to run?	N/A, leveraging AWS provided resources.
3. What is the demand or usage pattern for your service?	The prototype is for a small sample set of users of up to 50 concurrent operations.
4. What happens to your service when it experiences a surge in traffic or load?	Continuous Monitoring mechanisms alert when thresholds are reached, including CPU. For the prototype, we have not set up automated surge support, including spawning additional VM's, given our sample size for the prototype is a set of users up to 50 concurrent operations.

5. How much capacity is available in your hosting environment?	Given AWS model, we have the ability to add as much capacity as needed, on demand. The current server is an AWS EC2 instance, type M4 xLarge, with 4vCPU's, 16 GB Memory, and 50 GB Hard Disk.
6. How long does it take you to provision a new resource, like an application server?	On demand.
7. How have you designed your service to scale based on demand?	This is a prototype and has not been designed to scale yet.
8. How are you paying for your hosting infrastructure (e.g., by the minute, hourly, daily, monthly, fixed)?	Monthly fee based on utilization.
9. Is your service hosted in multiple regions, availability zones, or data centers?	This is a prototype hosted only in AWS US-East Data Center.
10. In the event of a catastrophic disaster to a datacenter, how long will it take to have the service operational?	This is a prototype.
11. What would be the impact of a prolonged downtime window?	N/A, there are only limited users because this is a prototype.
12. What data redundancy do you have built into the system, and what would be the impact of a catastrophic data loss?	N/A, EZRxLookup is not the source of the data.
13. How often do you need to contact a person from your hosting provider to get resources or to fix an issue?	N/A, this is a prototype.

Play 10: Automate testing and deployments

Checklist

1. Create automated tests that verify all user-facing functionality	x
2. Create unit and integration tests to verify modules and components	x
3. Run tests automatically as part of the build process	x
4. Perform deployments automatically with deployment scripts, continuous delivery services, or similar techniques	x
5. Conduct load and performance tests at regular intervals, including before public launch	x

Key Questions	Evidence
1. What percentage of the code base is covered by automated tests?	100% of all critical business flows and integration points are covered as identified in Acceptance Criteria. 50% line code coverage in general.
2. How long does it take to build, test, and deploy a typical bug fix?	Our build, test, and deploy process takes less than five (5) minutes on average.
3. How long does it take to build, test, and deploy a new feature into production?	On demand, but minimum of one (1) day.
4. How frequently are builds created?	Minimum of two (2) per day.
5. What test tools are used?	Junit, JMeter, and Selenium.
6. Which deployment automation or continuous integration tools are used?	Jenkins and Gradle.
7. What is the estimated maximum number of concurrent users who will want to use the system?	Prototype was tested to support 50 concurrent operations.
8. How many simultaneous users could the system handle, according to the most recent capacity test?	Prototype is being built to support a limited number of concurrent users. Our current performance tests are based on a single, medium sized Virtual Machine (VM) with 50 concurrent operations.
9. How does the service perform when you exceed the expected target usage volume? Does it degrade gracefully or catastrophically?	For the prototype, automated alerts and email notifications are configured to determine when thresholds are exceeded, but no automated recovery mechanisms are in place now.
10. What is your scaling strategy when demand increases suddenly?	For the prototype, automated alerts and email notifications are configured to determine when thresholds are exceeded, but no automated scaling is in place now.
Play 11: Manage security and privacy through reusable processes	
Checklist	

1. 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	N/A
2. 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	N/A; not source of data.
3. 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	N/A
4. Consider whether the user should be able to access, delete, or remove their information from the service	N/A
5. "Pre-certify" the hosting infrastructure used for the project using FedRAMP	N/A
6. Use deployment scripts to ensure configuration of production environment remains consistent and controllable	x
Key Questions	Evidence
1. Does the service collect personal information from the user? How is the user notified of this collection?	No
2. Does it collect more information than necessary? Could the data be used in ways an average user wouldn't expect?	No
3. How does a user access, correct, delete, or remove personal information?	N/A
4. Will any of the personal information stored in the system be shared with other services, people, or partners?	N/A
5. How and how often is the service tested for security vulnerabilities?	Daily with Open Web Application Security Project Zed Attack Proxy (OWASP ZAP).
6. How can someone from the public report a security issue?	This is a prototype and user feedback is collected via

	Stakeholder demos and usability tests.
Play 12: Use data to drive decisions	
Checklist	
1. Monitor system-level resource utilization in real time	x
2. Monitor system performance in real-time (e.g. response time, latency, throughput, and error rates)	x
3. Ensure monitoring can measure median, 95th percentile, and 98th percentile performance	x
4. Create automated alerts based on this monitoring	x
5. Track concurrent users in real-time, and monitor user behaviors in the aggregate to determine how well the service meets user needs	x
6. Publish metrics internally	x
7. Publish metrics externally	x
8. Use an experimentation tool that supports multivariate testing in production	N/A, this is a Prototype.
Key Questions	Evidence
1. What are the key metrics for the service?	Availability, Capacity, and Response Time.
2. How have these metrics performed over the life of the service?	N/A, this is a Prototype.
3. Which system monitoring tools are in place?	Amazon Web Services (AWS) provides infrastructure monitoring and ELK Stack provides Application monitoring.
4. 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?	The targeted response time is less than one (1) second 97% of the time and less than two (2) seconds 99% of the time. Current performance tests have 100% of all requests taking less than one (1) second.

5. 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?	During prototype testing, 100% of all requests, including during performance testing, are taking less than one (1) second.
6. What is the volume of each of your service's top 10 transactions? What is the percentage of transactions started vs. completed?	During prototype testing, including performance tests with 50 concurrent operations, we reached volumes in the thousands for our top transactions. During testing, all transactions that were appropriately started successfully completed.
7. What is your service's monthly uptime target?	This is a prototype, but our target uptime is 99.99%
8. What is your service's monthly uptime percentage, including scheduled maintenance? Excluding scheduled maintenance?	This is a prototype. Monthly uptime percentages not available.
9. How does your team receive automated alerts when incidents occur?	Via email.
10. How does your team respond to incidents? What is your post-mortem process?	The team adheres to IT Infrastructure Library (ITIL) incident, problem, and change management processes.
11. Which tools are in place to measure user behavior?	Logstash, Elasticsearch, and Kibana.
12. What tools or technologies are used for A/B testing?	A/B testing was not supported for the prototype.
13. How do you measure customer satisfaction?	Customer satisfaction is being measured with customer feedback forms for the prototype.

Play 13: Default to open

Checklist

1. Offer users a mechanism to report bugs and issues, and be responsive to these reports	x
2. Provide datasets to the public, in their entirety, through bulk downloads and APIs (application programming interfaces)	N/A
3. Ensure that data from the service is explicitly in the public domain, and that rights are waived globally via an international public	N/A

domain dedication, such as the “Creative Commons Zero” waiver	
4. Catalog data in the agency’s enterprise data inventory and add any public datasets to the agency’s public data listing	N/A
5. 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	N/A
6. 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	x
7. When appropriate, create an API for third parties and internal users to interact with the service directly	x
8. When appropriate, publish source code of projects or components online	x
9. When appropriate, share your development process and progress publicly	x
Key Questions	Evidence
1. How are you collecting user feedback for bugs and issues?	Feedback for bugs and issues are collected via the User Feedback Forms and Usability Test Reports.
2. If there is an API, what capabilities does it provide? Who uses it? How is it documented?	This application consumes openFDA APIs.
3. If the codebase has not been released under an open source license, explain why.	N/A
4. What components are made available to the public as open source?	All
5. What datasets are made available to the public?	N/A