****

**Requirement Specification for Drug Analytics**

Release 0.1

**Version**: 1.0

**Last Modified:** June 22, 2015

Prepared by:

TurningPoint Global Solutions

1355 Piccard Drive, Suite 250

Rockville, MD. 20850

**REVISION HISTORY**

| Version | Date | Organization/Point of Contact | Description of Changes |
| --- | --- | --- | --- |
| 0.1 | 06/22/2015 | TurningPoint DevelopmentTeam | Initial Draft Version |
|  |  |  |  |

# Overview

This document specifies the requirements for development of dAnalytics a prototype developed in response to the General Services Administration (GSA) Agile Delivery Services Request for Quotation - 4QTFHS150004. The requirements are from Open FDA open challenges Option 1[[1]](#footnote-1). The requirements are based on U.S Digital Services Playbook[[2]](#footnote-2).

# Business Requirements

1. **Level 1: Identify it.** Find a spike for a given drug query in the Adverse Events dataset and attempt explain it. For example, was there a recall or an enforcement report issued? Try bucketing by the following variables over time: weight, gender, or drug pairs (further broken down by drug characterization).
2. **Level 2: Normalize it.** Using publicly-available health-related data (medical care claims, discharge data, emergency room data) as a normalization method —  how does the spike in the adverse event series change, if at all?
3. **Level 3: Automate it.** Is there an algorithm that could be used to automatically identify such spikes?

# Technical Requirements

1. Consider Open source software solutions at every layer of the technology stack.
2. Choose software frameworks that are commonly used by private-sector companies creating similar services
3. Whenever possible, ensure that software can be deployed on a variety of commodity hardware types
4. Publish a repository consisting of all prototype source code, design assets, and all associated documentation that went into the creation of the prototype, to an online and publicly accessible version control system (e.g., GitHub, GitLab, BitBucket) that supports git. The uploaded repository shall be in git.
5. Create automated tests that verify all user-facing functionality.
6. Create unit and integration tests to verify modules and components.
7. Run tests automatically as part of the build process.
8. Perform deployments automatically with deployment scripts, continuous delivery services, or similar techniques.
9. Provide datasets to the public, in their entirety, through bulk downloads and APIs (application programming interfaces).

# User Interface Requirements

1. Create or use an existing, simple, and flexible design style guide for the service.
2. Use the design style guide consistently for related digital services.
3. Give users clear information about where they are in each step of the process
4. Follow accessibility best practices to ensure all people can use the service.
5. Use language that is familiar to the user and easy to understand.

# System Requirements

1. Resources are provisioned on demand.
2. Resources scale based on real-time user demand.
3. Resources are provisioned through an API.
4. Application is hosted on commodity hardware.
5. Monitor system-level resource utilization in real time.
6. Monitor system performance in real-time (e.g. response time, latency, throughput, and error rates.
7. Create automated alerts based on this monitoring.
8. Track concurrent users in real-time, and monitor user behaviors in the aggregate to determine how well the service meets user needs.
9. Publish metrics internally.
10. Publish metrics externally.

1. https://open.fda.gov/update/an-open-challenge-to-tap-public-data/ [↑](#footnote-ref-1)
2. https://playbook.cio.gov/ [↑](#footnote-ref-2)