****

**Drug Analytics**

**Development Approach**

**Version**: 1.0

**Last Modified:** July 7, 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 DevOps Team | Initial Draft Version |
| 1.0 | 07/01/2015 | TurningPoint DevOps Team | Modified based on comments. |

**Drug Analytics Development Approach**

[http://danalytics.tpgsi.com](http://danalytics.tpgsi.com/)

TurningPoint Global Solutions is an independently appraised CMMI Level 3 software development firm that has leveraged Agile Software Development best practices since 2005 to deliver user centric solutions for Big Data Analytics, Fraud, Waste, and Abuse, and Mobility to leading health agencies including FDA, CMS, DHA, VA, and HRSA. One agile-developed, user centric system we built has been used in 6 cabinet level agencies and is currently in sprint 56 (6 week sprints). Another solution we developed was recently named the 2015 winner of the Federal Health IT (FHIT) Innovation award in the Big Data category.

Our proven approach for designing, building, and operating effective, user centric digital services, which we tailor for each project to meet specific needs, is customized to government, and embraces checks and balances built within the Federal Enterprise Architecture (FEA), while still maintaining agility and emphasizing working software over comprehensive documentation. The paragraphs below describe the tailored approach that our team leveraged in response to the Agile Delivery Services RFQ to design and build the dAnalytics prototype (Link: [http://danalytics.tpgsi.com](http://danalytics.tpgsi.com/)).

Upon project initiation, TurningPoint assembled a small, focused team that included a Product Manager, Architect, Requirements analysts and DevOps engineers. The Product Owner (Project Manager) was given complete and final authority over the scope of the project and features to be implemented, and was ultimately responsible for how well the solution meets the needs of its users. His responsibilities included ensuring that features were built correctly, managing feature and bug backlogs, as well as budget management. Each team member was selected based on their experience in modern digital services, Big Data Analytics technology stack, and DevOps techniques, such as continuous integration and continuous deployment. The team was divided into a Requirements team and a DevOps team, which was then subdivided into Rest API, Hadoop and UI teams. To maximize productivity, team members were split between two shifts.

To ensure the success of the prototype we began the project by exploring and pinpointing the needs of “real people,” who would use the service. By incorporating likely end-users of dAnalytics in the design process from start to finish, we made sure that all technical and design decisions were informed by an understanding of what consumers of the prototype would actually need.

Based on discussions with the end users and research we performed to better understand the data and user community, we learned that patients needed a consumer friendly mechanism to understand adverse events that may be associated with drugs they have been prescribed. Data surrounding adverse events and spikes was publically available but had not been presented in a way that made it easily consumable or usable by the general public patient population. As a result, we developed requirements that were based on a set of hypotheses, which we then tested with these users. Some of the hypotheses were found to be false and we had to pivot and iteratively refine them.

We employed Agile development to reduce risk and allow working software to be delivered as soon as possible. By leveraging Agile development, we also handled task management more efficiently and with fewer handoffs. The DevOps team continuously integrated, tested and deployed every day to verify application performance. To proactively track risks and progress, and ensure communication between shifts we held twice-a-day standup meetings with a hand off to the second shift. During the twice daily standup meetings, we

* Reviewed new development as a part of the deployment, and
* Prioritized features to be developed or modified for the next day.

Code was developed outside of the master code base, to reduce risks of defects or bugs by enabling a code review before merge to the master code base. We added features iteratively throughout the development period to deliver an innovative, scalable, mobile and desktop digital service.

Our dAnalytics prototype is a publically accessible portal that consumes, modifies, remixes, and displays the Drug Adverse Event dataset. The prototype identifies spikes in the drug adverse event reported dataset and correlates this to any drug recalls. dAnalytics also allows users to filter data by weight, gender, and age so that they can have a more personalized and meaningful user experience.

We developed dAnalytics to work in an ‘Online’ and an ‘Offline’ mode to increase availability of the prototype and enhance the flexibility of the prototype for future iterations. In the online mode, the OpenFDA API data is queried, while in Offline mode, we query the same data which has been downloaded in our database. The offline mode enables dAnalytics to run independently of the OpenFDA API, making it available even when the OpenFDA API is offline. The offline mode also gives us the flexibility to develop new APIs in the future using the downloaded data.

We used modern, platform independent, open source technologies to reduce cost and enhance flexibility and interoperability. We used GitHub as the source code repository, for documenting user stories and defect tracking. Our technology stack also included Java 1.8, Apache Hadoop, MariaDB, Spring Boot, Angular JS, jQuery, Twitter Bootstrap, Apache Sqoop and Gradle. For the complete technology stack and architecture please refer to the [dAnalytics design document](https://github.com/TPRockville/dAnalytics/blob/master/Engg/Design/Documents/dAnalytics-Design%20document%20.docx). Some of the benefits for our selection of technologies include:

* Ability to process large datasets in a distributed environment, enabled by Apache Hadoop, which is used in the first step to preprocess the downloaded data.
* Rapid data retrieval. Apache Sqoop was used to load the data from Hadoop into MariaDB – a star schema relational database, which enables rapid data retrieval. MariaDB is a modern, open source relational database management system (RDBMS) derived from MySql. We have developed REST APIs to present the data from the MariaDB. The complete list of APIs developed is available in the dAnalytics design document.
* Responsive, user friendly user interface. dAnalytics web interface was built on AngularJS and uses REST APIs to render the data on the web page. We used AngularJS as it is a modern, open source web framework ideal for developing modern user friendly web applications. We used the cross platform jQuery library for plotting the graphs and charts on the screen. We have used Twitter Bootstrap to make dAnalytics user interface responsive and usable across desktop browsers and mobile devices. The UI prototype was refined iteratively with People to ensure that their requirements were addressed. The screen evolution is captured in meeting minutes, which are provided in GitHub repository.

The Business Analyst developed the algorithm for identifying the spikes in the adverse event data set. The algorithm was fine-tuned by running it on different data sets. The details of the algorithm are in [SpikeAlgorithm.docx](https://github.com/TPRockville/dAnalytics/blob/master/Req/Functional%20Spec/SpikeAlgorithm.docx).

To meet the accessibility needs of all potential users, we designed and developed the interface to be Section 508 compliant. The final UI was ‘Human Centric’ using end user interviews, wire framing and prototyping, simple, intuitive, and renders on both mobile and desktop browsers.

We ‘remixed’ the Adverse Events dataset with the Enforcement reports available from the APIs on opn.fda.gov. We query the enforcement API with the drug name to get any recalls issued by FDA for this drug.

To ensure the system met spikes in traffic and user demand, we deployed dAnalytics using IaaS provided by Amazon Web Services (AWS). We used Docker as container for the code deployment to achieve operating system level virtualization. The deployment architecture diagram is available in [dAnalytics deployment document](https://github.com/TPRockville/dAnalytics/blob/master/Engg/Release/dAnalytics-deployment%20document.docx). We monitor system health in real time using Monit. Monit is an open source tool that monitors CPU, memory usage and process status and is configured to send out alerts to an email distribution list. We designed dAnalytics with non-limiting architecture and AWS resource scaling to deliver a high availability, high performing solution.