VA Microconsulting Work Statement

DevOps Dashboard and API #60

March 6, 2020

POINT OF CONTACT:

Travis Sorensen
CEO, Oddball, Inc.
Phone: (301) 237-8505
Email: travis@oddball.io

Cage Code: 7HL75

DUNS Number: 080051825

Table of Contents

1.0 Introduction

To assist VA and OI&T in developing a strategy to create a self-servicing DevOps Maturity Model Dashboard environment, Oddball proposed a 3-phased approach that utilized an accelerated version of our Human-Centered Design (HCD) methodology. We designed our proposed method to focus on user and stakeholder research to discover VA's unique needs as they relate to DevOps environments and dashboards. This document contains a summary of results from the Discovery and User Research phases, as well as the key strategic recommendations distilled from the process.

As outlined in the original work statement (see Appendix A), our strategy assumed the existence of a dedicated team of seven to ten members that are responsible for the development and maintenance of the maturity model dashboard, referred to as the "Dashboard Team" throughout this document.

2.0 Results from Discovery Phase

During the first two weeks of the project, we initiated a kick-off meeting with the key VA stakeholders involved in the project. We used this time to develop a deeper understanding of the core problem explained in the original work statement (See Appendix A). We also dove into learning more about the categories of people (personas) that are relevant to proposing a strategic solution to that problem. We used this conversation and the information provided as part of the proposal process to generate a problem statement and to define user personas.

2.1 Problem Statement

The key problem was well-defined in the original work statement. All DevOps teams will be required to have:

> *"a dashboard and an Application Program Interface (API) that provides standardized performance measurement and DevOps maturity metrics across the product line."*

This gives the VA the ability to assess the maturity of their portfolio and individual products to help identify trends and help products and teams progress up through the maturity model.

2.2 User Personas

We determined there are three distinct user personas that should be addressed in our strategy:

- **Top Level Leader**
  This persona is characterized by the ability to assess VA DevOps as a whole with a maturity model that includes clearly defined facets. The responsibilities of this persona include tracking group performance and progress through the maturity model.
- **DevOps Manager**
  This persona needs the ability to assess their portfolio and products to identify areas and/or teams

that are not progressing up the model. They also desire to identify teams that are excelling and utilizing their process to make improvements in other areas.

- **DevOps Engineer**
  This persona needs the ability to drill down into detailed metrics data to understand their maturity metrics and detect and fix deviations. This persona is also looking for trends to identify areas of their work that are heading in a negative direction to take corrective action to improve their overall performance as a team.

We used these personas to identify candidates for the User Research phase of our HCD process.

2.3 Metrics for Success

We determined that there are several key metrics for success that should be addressed by our strategy:

- **Automated**
  The data should be able to be pulled into this dashboard in its end state. At first, it will likely be a manual process, but over time data sources should be connected to an API to be able to pull metrics onto the dashboard for all levels to examine.
- **Expandable**
  The ability of the dashboard to grow with the changing needs of the VA maturity model.
- **Multi-use**
  Allow different user personas to be able to utilize and assist them in determining areas of success and improvement.
- **Maintainable**
  The dashboard should be easily serviceable with management from the Dashboard Team as well as contractor-supplied and open-source contributions.

3.0 Key Points from User Research Phase

During the user research phase, we interviewed a number of individuals at different levels and relationships to the VA to help understand what their needs, solutions, concerns, and preferences would be for an automated DevOps dashboard and API. These interviewees came from several different levels of the VA DevOps organization, internal DevOps staff, and external DevOps contractors working in the private industry.

There were a number of findings from this research that directly contributed to our proposed strategy:

- **A high-level view of team maturity levels**
  There is no need to centrally collect and store *every* metric (some teams have thousands they already collect) but there is a need to clearly define and publish the data relevant for measuring the maturity metrics.
- **Preference for data to be exposed for collection by a central system**
  While most teams already both push some information (e.g. logs) and expose other information to be

pulled (e.g. existing Prometheus integrations), there was some preference towards exposing relevant metric data for pull by an external source.

- **Grafana and Prometheus are both widely used**
  Every team either used or was familiar with these two open-source metrics and visualization tools.
- **It is critical to support both modern and legacy systems**
  Like many large enterprises and organizations, there is a wide range of tools of differing ages and vintages.
- **Some metrics might need manual collection**
  Teams, particularly those at lower current levels of maturity or restricted by complex security regimes, may not be able to programmatically collect and distribute some of their metrics.

The interviews that led to these key points were transcribed and are included as part of the additional material delivered with this document.

4.0 Strategic Recommendations

We recommend a crawl-walk-run strategy for creating a self-servicing CI/CD DevOps environment that integrates with a Dashboard and is accessible via an API. This approach will leverage a small Dashboard team to provide self-service tools and infrastructure to external DevOps teams that can bootstrap their reporting process and leverage their existing tools and experience.

4.1 Tools Selection

The original work statement specified a “fully integrated, self-servicing, and expandable” environment, which we used to frame decisions around specific tool selection. Our interview subjects all mentioned their teams used the open-source tools **Prometheus** and **Grafana** as part of their internal monitoring and reporting, and we expect these tools will form the core of the technology platform for the DevOps Dashboard and API. These two tools have been incorporated into different programs at the VA already, such as the Lighthouse and VSP contracts. The collective expertise in the VA across teams and contracting companies, as well as the broad industry usage of these tools, makes it a clear and compelling choice. While we are aware that the VA has acquired tools at the enterprise level that may be strong alternatives to consider for filling in the data collection and/or dashboard publication roles, our time box and other constraints limited our evaluation of all the available tools. A breakdown of tools evaluated can be found in our supplemental PowerPoint deliverable.

4.2 Dashboard Team Implementation

The Dashboard team should be responsible for setting up and managing the central metrics repository (Prometheus) and dashboard presentation tool (Grafana). They will ensure that standardized metrics are collected from registered endpoints provided by each DevOps team. The dashboards should be designed for optimal clarity for the key User Personas and provide support through each step of the crawl-walk-run implementation for DevOps teams. We expect a small team can quickly provide this service.

These dashboard templates are created and managed centrally and deployed as part of the onboarding process for the DevOps teams. Each dashboard element will be tied to a key performance indicator from the VA Maturity

DevOps maturity model, which ensures a consistent and repeatable overview for each team. In the later stages of this process, a tool kit is provided (and managed) to assist in the speedy implementation and accuracy of the data.

4.3 DevOps Team Implementation

We anticipate that many teams can quickly move into the Walk step, but realize that there are issues around tool experience, operational security, and team maturity that could result in some teams starting and remaining in the Crawl step for some time. We recommend identifying a small group of candidate teams to pilot the implementation process before a general rollout.

4.3.1 Crawl step

In order to get a DevOps team up and running with actionable data quickly and simply, the Dashboard team will use a Grafana template to create the dashboard and collect data via surveys. Surveys will be sent on a regular interval (such as monthly or with each sprint) to keep the dashboard information fresh until teams are ready for the “walk step.”

The metrics from the surveys are then added to the central repository using scripts built by the Dashboard Team. For example, a Prometheus repository could be integrating using the Push Gateway which is designed for "service level" batch jobs which directly matches a regular, structured collection of arbitrary data from a DevOps team. Automating data collection from a web-based form engine should be a one-time effort of the Dashboard team that will automate the collection of the survey results.

4.3.2 Walk step

During this step, the DevOps teams will transition from the survey-based metrics to exposing preconfigured metric endpoints that are supplied by the Dashboard team in the form of a tool kit preconfigured for Prometheus consumption. They can then register the metric endpoints with the central metrics repository, and work to ensure that the metrics are flowing correctly to the central repository. Once the endpoints are registered, the central organization can begin pulling the data at that endpoint and reporting the maturity data in the team dashboard at the same cadence as the survey. The endpoints initially contain the preconfigured baseline values from the central team, and the DevOps team would be responsible for updating this value as they mature. As the team transitions to “running,” these values will be updated programmatically by the team’s monitoring tools.

As maturity model definitions change over time, this stage provides a repeatable and scalable way to add additional or different statistics. The Dashboard Team would need to build and provide a default template for the new JSON response that each team can add to their existing deployment and then update the dashboards to reflect the latest statistics. Since each DevOps team will have been through this process already, the additional work should be straightforward and minimal.

4.3.3 Run step

Once DevOps teams are utilizing the provided tool kit and providing default JSON data to the Dashboard team, and dashboards are being successfully populated, they can begin to use their existing monitoring tools to begin serving live, regularly updated data to the registered endpoints so that the dashboards are dynamically

updated. We recommend that the Dashboard team clearly define the polling frequencies so that DevOps teams align their data collection and publication process appropriately.

We expect that many DevOps teams are likely to be collecting some or all of the required data for their purposes, which will accelerate the rollout of this step. Metrics with existing sources will likely be integrated quickly. New or uncollected metrics will generally take longer to build into existing systems before being published to the central repository. In both cases, DevOps teams will be able to use their existing experience with existing tools to move forward quickly.

5.0 Next steps

We recommend that the VA trial this crawl-run-walk strategy with 2-3 teams that are currently at a mid-to-high level of maturity to test and verify the value of this strategy. We further recommend that the teams chosen have some expertise with Prometheus and Grafana to move the process forward as fast as possible. Once their candidate teams are chosen, each one should:

- Build a default Grafana dashboard based on the defined key performance indicators in the VA DevOps maturity model in consultation with the Dashboard team
- Create a tool kit for the corresponding default metric endpoints that expose those values
- Work with the Dashboard team to set up a demonstration Grafana and Prometheus environment

Once these steps are completed, the VA would be able to complete a pilot project with a small number of internal DevOps teams to ensure that this scales at the appropriate level of effort, is usable by multiple teams, and automatically collects the metrics for building the DevOps Maturity Dashboard

Appendix A

Restatement of proposal Deliverable:

A defined and repeatable strategy/process to create a fully integrated, self-servicing, and expandable DevOps environment that can be managed by a small team of seven to ten members, and contributed to by the open-source and contractor community.

Final Deliverable should be a White Paper / PowerPoint no longer than 4 pages that outlines:

- Defined Strategy / Process in written and PowerPoint format that details the how-to
- The Strategy / Process should outline how to stand up the Dashboard and identify the API Key Performance Measurements needed to assess DevOps Maturity