

Tracking Workflow Metrics: How I Used Design to Improve Efficiency

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Siddharth Tickle

Fall 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Rosanne Vrugtman

ABSTRACT

Qualtrics needed an easier method for tracking internal metrics since its current solution forced users to go through multiple third-party services to access the information. As a member of the xFlow Runtime team, I developed a web-app using React, JavaScript, Node, Jenkins, and Mocha to serve as a dashboard and a one-stop shop to view the data from each of the third-party services. First, I identified the problem and areas for improvement by speaking to potential users, other engineers on the team, and my mentor and manager. I then created a design document and mockups for the web-app using Figma, a wireframing/mockup tool, to visualize the app at a low cost. Next, I presented the design document and proposed project to my team for feedback and changes. This cycle continued until I had perfected the plan for the project and was ready to start development. Some features have already been added, and the tool is still in use today. The next feature added will be an option to retry workflows directly from the tool rather than having to log in to the internal service.

1. INTRODUCTION

The problem revolved around debugging workflows, a product widely utilized within the company. Workflows, triggered every time a user submits a Qualtrics survey, can consist of a variety of tasks utilizing native and third-party services. Since the product is used within many engineering and customer service teams, they all need access to data related to specific workflows. To access JSON data about a workflow, users must go through multiple services. For example, to access an ID related to a specific workflow, users must go to the workflow within the Qualtrics service, copy the ID, go to the Amazon DynamoDB instance with that specific ID having permission from engineers on other teams,

then finally view the information associated with the workflow.

This problem was consistent with many different fields related to a workflow. The current solution was to use two existing APIs to access this information. However, this was not customer-friendly and there was no existing user interface for users to access the same information. This was important because the target users not only consisted of engineers, but also the resolutions team in charge of customer service and support calls. Thus, there was a need for a UI-based tool to retrieve data associated with workflows.

2. RELATED WORKS

Before beginning development of my project, I was tasked with researching existing tools that accomplished a similar goal within the internal library of software. The first tool I reviewed, was also a debugging tool, but was related to a different feature within the Qualtrics product space (Qualtrics, 2022).

This product also presented a user interface that allowed users to enter an ID in a field and retrieve data associated with the product. The main points of interest that influenced my project were the tech-stack used and the structure of the site. The technologies used included TypeScript, Go, and React. Although I did not use Go in my project, the modularity of TypeScript and React inspired me to take a similar approach in my project. I knew my time working would be limited and wanted my project to survive after I left the company, so I made it very easy to add new features to the project without making significant changes to the existing user interface.

The second resource I followed consisted of discussions with other members on my

overall organization. Everyone in the organization was actively invested in the project because they knew it would improve their workflow. Thus, presenting the project at office hours presented customer feedback from multiple viewpoints. An improvement I took away from the meeting was that my original understanding and usage of jargon related to the project was not correct, so I adjusted my designs to follow the process. Additionally, I was initially planning to use my own login authentication but was then prompted by many engineers on my team to use instead sectool, an authentication service developed by Qualtrics.

This meeting was necessary because it gave me a chance to speak with my target customer directly about the product. This is a vital stage in the development of any product and doing so here made development at later stages much easier.

3. PROJECT DESIGN

This section will cover the system architecture, what services were used, and requirements set in place for the project before it began.

3.1 Review of System Architecture

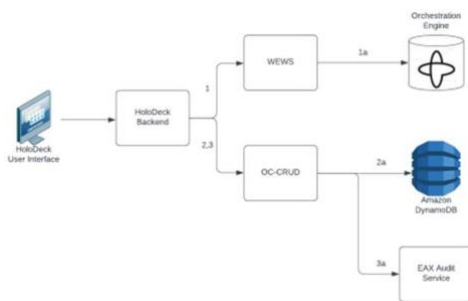


Figure 1. System Architecture

The project, named HoloDeck, used two different services to serve as its main architecture, as seen in Figure 1. The first service it uses is the WEWS service to retrieve JSON data regarding workflows. The second service hosts two endpoints,

one to retrieve the revision history for a given workflow and another to enable or disable a workflow.

3.2 Requirements

For the scope of this project there were multiple requirements. The first three fall in line with the system architecture. Users should be able to retrieve workflow data, workflow revision history, and enable/disable a workflow from the project. In addition to these three users had to be able to log in to the webapp with the secure service used by Qualtrics at the time.

3.3 Key Components

The webapp utilized React since it is allowing for a modular approach to development. The webapp consisted of three main components with more space for improvement in the future. The current three components followed the same format as the system architecture. There was a component to view workflow data, one to view revision history of a given workflow, and one to toggle a workflow on or off.

4. RESULTS

The HoloDeck improved the time needed to conduct simple tasks on a workflow. Before the HoloDeck users had to log in to a variety of different internal logs and mechanisms to access data to aid in debugging workflows. Both Qualtrics engineers and Resolution team members benefited from this webapp as it reduced the amount of time needed to access a workflow, its revision history, or its enabled status by a significant percentage.

5. CONCLUSION

The HoloDeck lived past the length of my internship and served an important role by improving the efficiency of debugging workflows. Before the end of my time at

Qualtrics, other engineers already began using the tool and came up with more suggestion for potential improvements in future versions of the software. Since the software was built to be modular, it has multiple different pathways to take. One engineer has already added new features based on this modular architecture by using the documentation I authored.

6. FUTURE WORK

The HoloDeck was built upon using the modular architecture it was developed with. In the future, more features will be added similarly to the way a feature to retry workflows was used. The HoloDeck has a lot of potential for improvements from the way it was built. Since we used React, a new component can be created and rendered to a new tab whenever a new feature is desired. A continuous cycle of customer feedback is also in play since the developers who are now working on the HoloDeck are also using it to debug the workflows they are using in other projects. Thus, the HoloDeck will always have a constant stream of feature additions.

7. UVA EVALUATION

Most of my prior education in the technologies I used for my project (JavaScript, React, Node, Figma, etc.) came from my own interests and work outside of school. I feel as though UVA students would benefit from a class that provides students with an education on modern technologies rather than less common technologies like Django or PHP which are currently offered. As compared to other institutions, UVA lacks course offerings in these modern technologies that would prepare students for careers in software development/engineering, and when it does the classes are too small and fill up too quickly by the time enrollment comes around.

8. ACKNOWLEDGMENTS

My mentor during work, CJ Kim, was extremely helpful not only technically, but also socially. He never failed to answer my questions and wanted to see me succeed as a software engineer.

REFERENCES

Kim, C. J., & Bodin, K. (2005). Retrieved 2023, from <https://www.qualtrics.com>