
Aetna DevOps Dashboard

Team MMAC

Max Meyer, Matt Rusczyk, Austin Barrett,
Cody DallaValle

— Itinerary

- Meet the Team
- Project Overview
- Our Experiences
- Our Environment
- Demo
- What Next?
- Conclusion

Meet Team MMAC

Max Meyer



- Scrum Master
- Team Contact
- Test Data Creator

Max is a senior at CCSU pursuing a Bachelors of Science degree in the Computer Science Honors program. He has obtained a working knowledge of the following programming languages including C, Java, and SQL.

Max first became interested in coding while taking a MATLAB course in the Mechanical Engineering program at CCSU.

As a relatively new student to the CS Major, the Fall 2017 semester was a crash course in advanced CS courses for Max. Working on team projects which consisted of a Self-Watering plant IoT device for Systems Programming, as well as an MP3 player in Design Patterns has broadened his scope as a CS student.

Matt Rusczyk



- Angular Converter
- AWS Administrator
- Accordion Collapser

Matt Rusczyk is a senior at CCSU in the Computer Science program. His programming experience is in the following languages Java, R, Hadoop, MySQL, & C.

When Matt was a Mechanical Engineering Major, his favorite courses were Finite Elemental Analysis (FEA) and Manufacturing Processing. To complete his Big Data he built a HDFS cluster in his bedroom.

Matt enjoys golfing, bike riding, and attempting to create smart devices with Raspberry Pis.

Austin Barrett



- Git Master
- SignalR Whisperer
- Load Time Minimizer

Austin Barrett is a senior pursuing a double major in Computer Science and Mathematics at CCSU. His programming experience is in C#, Java, C, C++, JavaScript, and Swift.

Although he had already been writing simple programs for years, Austin's first foray into programming was by writing TI-BASIC programs on his TI-84 to avoid doing homework during his senior year of high school.

This summer, Austin began developing an ASP.NET password manager for internal use at the credit union he was working at, using C# for the backend and PostgreSQL for the database.

Cody Dallavalle



- API Mastermind
- Graph Visionary
- Modal Modeler

Cody DallaValle is a fifth year student at CCSU majoring in computer science and minoring in mathematics.

Cody has experience programming with languages such as C#, Python, LISP, Javascript, ANSI C/C++, and Java. His first experience coding was with Lua scripting for computer games and later VB.NET for Windows applications.

After graduating, he hopes to work as a software engineer for several years and research machine learning. An example of real world software Cody created was a Chrome extension designed to automate ticket creation from CSV files which were exported from large Excel spreadsheets.

The Dashboard

0. Project Goals

- To provide a means of combining output from different aspects in the DevOps product lifecycle.
- Display the data in means both intuitively laid out as well as informative from a glance

A hand holding a smartphone, with the screen displaying a dashboard. The background is a blurred image of a person in a red shirt.

**The dashboard is
designed to be used
on desktops and
mobile phones**

1. Intro

This application was developed by Web Engineering/Platform Engineering in collaboration with Team MMAC, utilizing the following technologies:

- **AngularJS**
- **Bootstrap**
- **ASP.NET Web API**
- **Octopus API**
- **ChartJS**
- **TimelineJS**
- **MomentJS**

2. Data Collected

Number and name of:

- Projects
- Project Groups
- Lifecycles
- Machines
- Environments

Deployment events:

- Started Deployments
- Queued Deployments
- Successful Deployments
- Failed Deployments

3. Visualizations

- Line and Bar graphs, Pie charts
- Collapsibles to initially hide more detailed information
- Timeline detailing the releases and deployments of projects
- Bootstrap Modals to focus attention

Our Experience



Obstacles

Software engineering is not easy. We ran into many problems along the way and we weren't able to solve every one:

- VS setup for full stack web development
- Learning curve for related technologies
- Team communication and cooperation
- Common git problems
- Dynamically creating timelines and links in modals
- SignalR
- Reloading graphs with new data
- Making the data relational
- Merge conflicts
- Datetime conversion between APIs and the frontend
- Mobile testing
- Angularizing things

Production Environment

Amazon Web Services was used to host Octopus Deploy
We Utilized:

- Three Amazon EC2 instances
 - ◆ Two Linux instances
 - Two deployment targets
 - One MSSQL instance
 - ◆ One Windows instance
 - Hosted Octopus Deploy application
 - Acted as Windows deployment target

Development Environment

- GitHub was used for source version control.
- Visual Studio 2017 was used to develop the dashboard.
- We have tested the dashboard on Chrome, Safari, and Firefox
- To test the dashboard on mobile, VS was configured to debug using IIS.
- NuGet packages and other scripts we added include angular-momentjs and angular-chartjs.
- AngularJS was used for making client-side calls to internal API and updating the UI components with the results
- Bootstrap was used for responsive page layout, styling, and UI components
- ASP.NET controls the backend, which is responsible for calling the Octopus API and sending the client updated info
- ChartJS was utilized to show rich visuals of the data
- MomentJS was used for converting date/time strings and showing human readable dates
- TimelineJS was used for representing the timeline of releases and deployments for each project

Demo

What next?

The dashboard is not perfect and can be improved in ways, some of the many things we didn't get to do:

- Relating the data more uniformly (the model classes currently don't have meaningful relationships because they're mostly just used to send data along)
- Fixing UI issues (hardcoded delays can be changed to events, replace normal elements with bootstrap)
- Pagination (right now the internal API only asks the Octopus API for a small number of results)
- Authentication (we generated an Octopus API key which is used on each call, but it could be done without this)
- Linking to TeamCity (currently the release notes for deployments has info relating to TeamCity builds but no way of accessing the TeamCity API)
- Filtering of projects and deployments (we thought the Environment list could be used to filter the deploy graphs and the Project Group list could filter the project list)

Conclusion

Team MMAC thanks Aetna for their time and this opportunity

Problem : Multiple DevOps environments results in a wealth of information

Solution: Octopus Dashboard brings deploy data and links to Octopus together in one concise place

Key Uses: Management gets access to data overview of relevant projects

With continued development the Dashboard looks to be a useful app for Aetna DevOps Employees