

**Assignment 8:
Individual Requirements Analysis
for Semester Project**

CHAOSS Augur

Samantha Sample

Introduction

With the growing popularity of and emphasis on open source technologies, developers and companies are now more invested than ever in creating rich and engaging projects. As these projects augment, it becomes more difficult to determine the overall health, and therefore viability, of a given repository. In order to foster sustainable projects, it is important to have active metrics that can be used to determine the overall health of a repository. These insights can then be used to inform meaningful changes in the maintenance of any repo.

Software Product Overview

For our product, we wish to contribute to an existing solution to this problem, Augur. According to their [Github](#):

“Augur is a prototyped implementation of the Linux Foundation's [CHAOSS](#) [Community Health Analytics Open Source Software] Project on [open source software metrics](#). Technically, Augur is a [Flask web application](#), [Python library](#) and [REST server](#) that presents metrics on open source software development project health and sustaCHAOSS inability.”

Overall, Augur is working to provide metrics on the health of open source software, enabling insight on how to improve it. These metrics enable helpful comparisons across projects and over time, allowing the results to reflect and confirm any positive changes in the organization/methodology of the project. With this data, it is easier than ever to create an open source community is positive, inclusive, and sustainable.

To contribute to this project, we would like to implement new metrics within the Augur Python library, along with new REST server endpoints for accessing this data. We will also create a website which will host visualizations of both new and existing endpoints to allow for easier interpretation of this data. Users can choose a repository or group of repositories, and view the available data dynamically, which can then be used to inform potential improvements in the project. This is helpful to current and potential project contributors, project managers, and the users of the product.

System Use

The system is intended to be used by people who are literate on software terminology and familiar with the fundamental aspects of OSS. Users will be able to

view the metrics as displayed on our webpage, or may utilize the newly created endpoints to access the raw data from each metric.

Actor Survey:

- 1) Project Contributors: These are the developers making technical contributions to a given repo or repo group, by means of commits, pull requests, issue creation, and more. They are most often the subjects of our metrics, as they are the key players in crafting and maintaining a given repository.
- 2) Project Managers: These are the individuals/companies planning, directing, and overseeing a given project. Though they may not be making contributions to the project repo, they are instrumental in shaping the project culture and have a large investment in the health of the project. They are key users of the metrics.
- 3) Project Stakeholders: These are the people/companies who use the products being analyzed. They are directly impacted by the overall quality of the project, and may even choose to view the metrics themselves to inform their decision to use a given product or not.
- 4) System Administrators: These are the people overseeing and maintaining the public augur metrics and site. They make sure the website is easily accessed and functioning.

System Requirements

Use Cases:

1)

Use Case Name	Endpoint Call
Description	The Augur REST API will return a JSON response containing relevant data for a specified metric
Primary Actor	End users
Secondary Actors	Project Contributors, Project Managers
Trigger	The end user makes a call to the REST API
Preconditions	The server is up and functioning properly
Postconditions	None
Scenario	An end user wishes to view data associated with a certain metric, for instance test coverage, on a specific

	repository or repository group. To view this data, a call must be made to the Augur REST API containing the repo/group id, desired metric, and any concatenating/formatting specifications. This call is passed over HTTP, and a response is generated through a corresponding Python function and SQL query to the Augur database. When the response has been formulated, it will be returned to the user as a JSON.
Alternate Flow	If there is no data associated with users request, an empty JSON will be returned.
Exception Flow	If an error occurs during the API call, an error message will be returned to the user asking they try again.
Extensions	None
Notes	None

2)

Use Case Name	Generate Website Visualization
Description	Our website will display graphs and charts using data from the Augur database for some metrics
Primary Actor	End user
Secondary Actors	Project Managers, Project Stakeholders
Trigger	A user will select a repo group and optionally a repo, clicking submit to generate the chart.
Preconditions	The website is up and functioning
Postconditions	The website will be ready to complete a new call
Scenario	When someone would like to view a specific graph on our website displaying data for a certain repo (group) and metric, they will first navigate to the website. Then, they will select the repo (group) they would like to see for any of the metrics available on the website. When they click submit, a call will be made to the Augur API for the data, which will then be used to popular the chart/graph for the user to view. If they choose, they can download the graph as an image.

Alternate Flow	If no data is available from the API call, a message will be displayed informing the user
Exception Flow	If an error occurs, a message will be displayed informing the user of the error
Extensions	None
Notes	None

Functional Specification:

- 1) A public website will be available where a user can choose a repo (group) and view visualizations of the data corresponding to a selection of metrics.
 - a) These visualizations will be dynamic and current, and will accurately reflect the available data.
- 2) A REST API will be available for developers to make API calls for each available metric, which will return corresponding data as a JSON
 - a) Each endpoint will return data in a standardized format which will be easily interpreted by the user.
 - b) Information on the use of the API and endpoints will be documented in the Augur Documentation for ease of use.

Non-functional Requirements:

- 1) All the metric visualizations should be understandable by at least by 75% of all viewers
- 2) All API calls should return in under 10 seconds.
- 3) The website will provide robust error messages for each time a repo (group) does not supply data for any graph.
- 4) The system must be available 99% of the time.
- 5) The system should support at least 10 simultaneous API calls without error.

Design Constraints

- 1) All new endpoints should function similarly to the existing endpoints, and should not alter or remove the functionality of any other endpoint.
- 2) The website and API must be secure and safe to run on most computers, and should not require any highly specific tools or hardware to operate.

Purchased Components

This project will need a server in order to host and run the application, as well as space to store all relevant data in a database.

Interfaces

The primary interface for this project will be a public website in which users can read and learn about the available metrics, as well as generate appropriate visualizations using those metrics for a particular repo or group. These dynamic data generation will be conducted using HTTP protocols over a public API. The database hosting this information will use PostgreSQL.