Sprint 1

Dee Archdekin, Nathanael Hagemeyer, Melissa Tully, Conor Wood

Endpoints:

1. Maintainer count
2. Organizational influence
3. Organizational Contributing (Joining or Leaving)
4. Organization or Volunteer Driven
5. Peripheral Organizations

Maintainer Count Endpoint

- This endpoint would create data for the “Maintainer Count” CHAOSS metric. The intended use of this metric would be to assess the number of individuals that are assigned the “maintainer” role for a GitHub repository. This role grants someone higher-level permissions that allow them to “maintain” a repository without access to sensitive information and should belong to an active, visible member. The existence of this endpoint would be useful to gather how many individuals have this role, and assess the activity of these individuals to ensure that they are fulfilling their duties as a maintainer. It could also be a useful metric to produce to ensure that there is an appropriate number of individuals that possess this role.

Organizational Influence Endpoint

- As an approximation of influence, this metric will indicate the percentage of commits by users who are part of an organization in a given period. For individuals or companies considering contributing to an open source project, knowing that one company dominates or that it is developed by unaffiliated individuals may inform their decision. This metric could be used for visualizations as well, either dividing the history of a project into equal segments and/or showing the cumulative breakdown.

Organizations Contributing (Joining or Leaving)

- This endpoint generates data for the “Organizational Contributing” metric. This metric attempts to measure the impact of organizations joining and leaving the project. It uses the affiliation start date and last contribution date for each contributor affiliation, as well as the commits per month before, during, and after the affiliation. This endpoint will allow evaluation of the changes in commit activity as organizations join and leave.

Organization or Volunteer Driven

- This endpoint would create data for the “Organization or Volunteer Driven” metric. This metric indicates the ratio of commits by users affiliated with an organization vs. users with no organizational affiliation.

Peripheral Organizations

- This endpoint would create data for the “Peripheral Organizations” metric. This would show the organizations who had a temporary influence on the project but have since stopped contributing. For individuals or organizations who are considering joining to contribute to a project. If they see that a lot of organizations made the same decision and then stopped, it might say something about the health of the project.

Component	**Requirement Name**	**Requirement Description**
All Endpoints	Access Database	Endpoint successfully connects to database
All Endpoints	Query Data	Endpoint executes query that returns desired data
All Endpoints	Return Data	Endpoint returns data formatted to conform to Augur API standards
All Endpoints	Standardization	All metrics are formatted to meet Augur API standards
Maintainer Count	Asses maintainer activity	Returns data such that the user can tell how many maintainers are active and fulfilling their duties vs. how many are not
Organizational Influence	Variable Time Period	Query returns organizational influence data over a given time period, up to the lifetime of the project
Organizations Contributing	Distinguish Active Organizations	Endpoint returns data formatted such that organizations that are still active can be distinguished from organizations that are not, and active organizations do not have commits per month after leaving
Organization vs. Volunteer Driven	Supports Pie Chart	Data returned allows generation of a pie chart visualization of which organizations have contributed which percentage of commits

Peripheral Organizations	Different timeframes	The endpoint returns data that tells the user how many organizations have dropped the project within different timeframes- eg. how many have dropped out in the last 3 months, 6 months, and year

Use Case

Title: Evaluate Organizational Impact via CHAOSS Metrics

Description

A variety of different user types want to analyze different aspects of a project. A user would identify the correct endpoint/set of endpoints to retrieve the data in order to analyze the desired aspects. When the user finds the endpoint(s) that they need, they can query the endpoints which will then perform operations to retrieve that data from a database with the repository previously loaded. The retrieved data will then be formatted and returned to the user for further analysis. In case of an error while retrieving the data, an error message will be returned back to the user instead. Once the user possesses the data, they can perform various operations that can be used for analysis of their project.

Triggers

1. User requests the desired data by querying the appropriate endpoint
2. Another service is calling the endpoint for use in a visualization

Actors

1. Project Manager
2. Repository Owner
3. Hobbyist

Preconditions

1. Specific repository has been selected for analysis
2. Data for repository exists in the database instance

Main Success Scenario (Goals)

1. All requested data is accurately returned to the user
2. The user gains insight about the different organizations that contributed to the repo

Alternate Success Scenarios

1. Partial data or improperly formatted data is returned and the user can use the provided data to analyze other portions of the project

Failed End Condition

1. Data for the desired repository is not found
2. The data is presented in a way so that the user learns nothing from it

Extensions

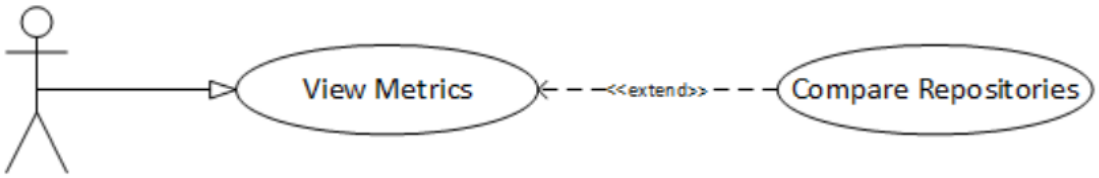
1. Compare Repositories

Steps of Execution (Requirements)

1. User identifies the repository and metrics for said repository they want to retrieve

2. User queries the endpoint(s)
3. Data is returned to user in JSON or other similar format
4. Users can utilize the returned data to perform various tasks such as creating visualizations, analyzing project metrics, etc.

A use case diagram



Dependent Use Cases

1. Visualize metric