Title: 1-org-outside

Background

Organizations evaluate open source communities before becoming engaged or using open source software to satisfy a need. Depending on the context of the need, organizations may want to assess the sustainability of the community. Inside of an organization, employees are the decision makers and have to justify that engaging with an open source community meets the needs of the organization and related guidelines are met.

Description

The user (employee) provides a URL to a GitHub repository for a community that they want to learn about. The software returns metrics that indicate the health and sustainability of the repository.

Triggers (What prompts the use case to start?)

1. A user (employee) seeks an open source solution to an organizational need and must assess the health and sustainability of an open source community before getting approval to engage or contribute.
2. An outside organization seeks to get metrics on an open source solution to determine if they want to continue to use it or being using it

Actors (Who is involved?)

1. Managers
2. Outside organizations
3. Employees

Preconditions (This includes things like “data loaded”. Or, project is flagged as “of interest”; etc.)

1. GitHub repository exists
2. User provides URL to a GitHub repository
3. User has access to view metrics of that repository

Main Success Scenario

1. All metrics that can be computed from the provided repository are displayed.

Alternate Success Scenarios

1. N/A

Failed End Condition

1. The provided URL points to a non-existent GitHub repository, metrics cannot be calculated, and an error message explaining the condition is shown to user.

2. The given repository does not have enough data for metrics to be computed.
3. The user does not have access to view the metrics on that repository

Extensions

1. Compare metrics between repositories.

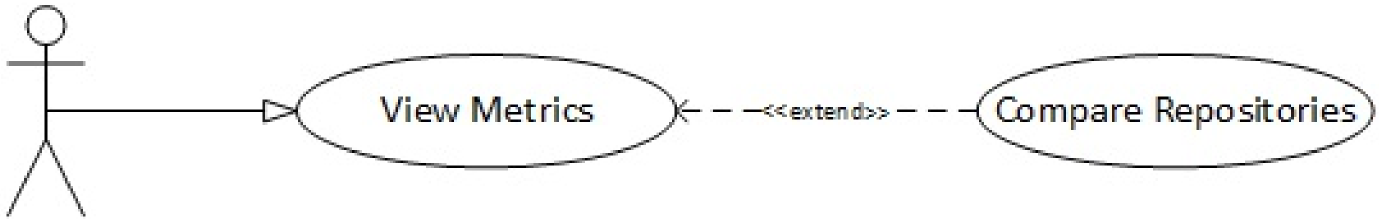
Steps of Execution (Requirements)

1. The user enters and submits the URL to a GitHub repository.
2. Metrics computed from the provided repository are displayed.

Dependent Use Cases

1. N/A

A use case diagram, following the UML Standard for expressing use cases.


View Metrics
<<extend>>
Compare Repositories