1. **Introduction**

Open source projects are becoming increasingly important in many industries so metrics tools to help monitor the health of such projects are becoming important as well. This document lays out the basic requirements for an open source metrics tool the monitors commit activity of a repo group. The goal is to track the commits and display the information in a way that shows activity trends of the group as well as of specific users. This will allow those using the tool to check how active a current project is, or if the amount of activity around the project has declined.

1. **Software product overview**

A product designed to display commit activity trends in a repo group. These trends will be displayed using the number of commits over a period of time. These activity trends will track user and community activity over time. This will allow contributors and interested parties to easily check the activity of a project. The goal of the project is to monitor when a project is starting to increase or decline in the amount of participation. At this point in time the scope of the project is limited to tracking only the number of commits over time.

1. **System Use**

This project really only has two use cases which will be openly available to all users. These are checking a repo group commit activity trends, and checking a user commit activity trends for specified repo group.

* 1. **Activity Survey**

**Contributor:** This defines any user that makes commits to the related project. A contributor is very likelyto have an active interest in the health of the project as they are one of the people responsible for its creation. They are also likely to want to check their own activity and thus will be likely to use the option to track user commit activity trends to track their own activity.

**Manager:** This defines people in a project who take on more of a management role in the project. Although not all open source project may have someone that fits this description, projects can often benefit by having one or more people that can act to direct the project. Manager would be particularly interested in checking commit activity trends to see what kind of progress is being made as well as to see which users are active.

**Sponsor:** This defines a person or entity that has invested in a project. Although many open source projects are group collaboration with little money involved, many larger projects may have funding of some kind. A Sponsor would, for obvious reasons, be interested in checking the activity of the groups they are funding. In addition, if particular users are on a payroll, the sponsor may wish to check the commit trends of those particular users as well.

**Other Interested Party:** This defines any party that is interested in a project, but doesn't fit the descriptions above. They could be a user of some portion of a project, someone doing research on a project, or just a curious person who happened to develop an interest in a project. These users would be less likely to care about specific user activity as they likely wouldn't know any of the users anyway. However, they may still care about the group's overall commit activity trends.

1. **System Requirements**

|  |  |
| --- | --- |
| Use-case name | U01 Check commit activity trend in repo group |
| System or subsystem | Main |
| Actors | All users |
| Brief Description | A user requests for the commit activity trends for a specified repo group. The request is sent to a server API when then returns the data. The data is then displayed on a web application in a readable fashion. |
| Basic flow of events | Basic flow begins when user selects a repo group, but leaves the user blank.   1. Data request will be sent to server 2. Data will be returned to user's application 3. Application will display data in readable fashion |
| Alternate flow of events | Error retrieving data.   1. Send back error 2. Display an error message to user |

|  |  |
| --- | --- |
| Use-case name | U02 Check commit activity trend for user in repo group |
| System or subsystem | Main |
| Actors | All users |
| Brief Description | A user requests for the commit activity trends for a specified user in a repo group. The request is sent to a server API when then returns the data. The data is then displayed on a web application in a readable fashion. |
| Basic flow of events | Basic flow begins when user selects a repo group and a user.   1. Data request will be sent to server 2. Data will be returned to user's application 3. Application will display data in readable fashion |
| Alternate flow of events | Error retrieving data.   1. Send back error 2. Display an error message to user |

**4.1 Functional Requirements:**

U01: Application will display repo group commit activity for select group.

U02: Application will display results of commit activity of selected user in selected repo group.

**4.2 Non-functional Requirements:**

General non-functional requirements: System will display proper data for all repo group and user options.

U01-1: Application will have line chart displaying results.

U01-2: Line chart will have number of commits as the vertical axis and date as the horizontal axis and the line will be label with the repo group name.

U02-1: Application will have line chart displaying results.

U02-2: Line chart will have number of commits as the vertical axis and date as the horizontal axis and the line will be label with the user name.

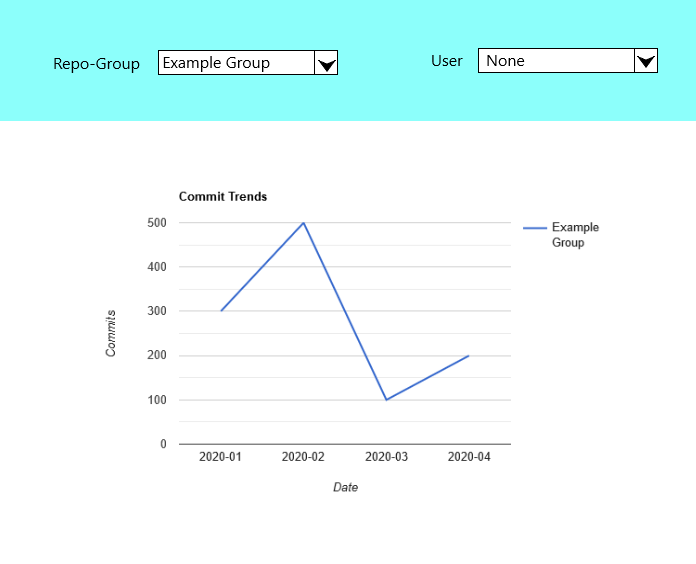
1. **Design Constraints:**

* web deployable
* available to any interested party
* easily readable and usable
* continuously updated
* crash resistant (will not break if an error occurs)
* accurate commit information (will not display incorrect values)

1. **Purchased Components:**

* A server to store and retrieve commits information. This could be a physical server or one rented from a services such as Amazon AWS.

1. **Interface for Repo-Group trend:**



Interface with User Trend:

