STARTER PROJECT HEALTH

Lihao Fan
Adam Routt
Josh Jaworski
Shane McElvey
Drew Rothweil

.

This project pulls from 2 metric models: Starter Project Health and Community Activity.

Community Activity contains the following metrics:

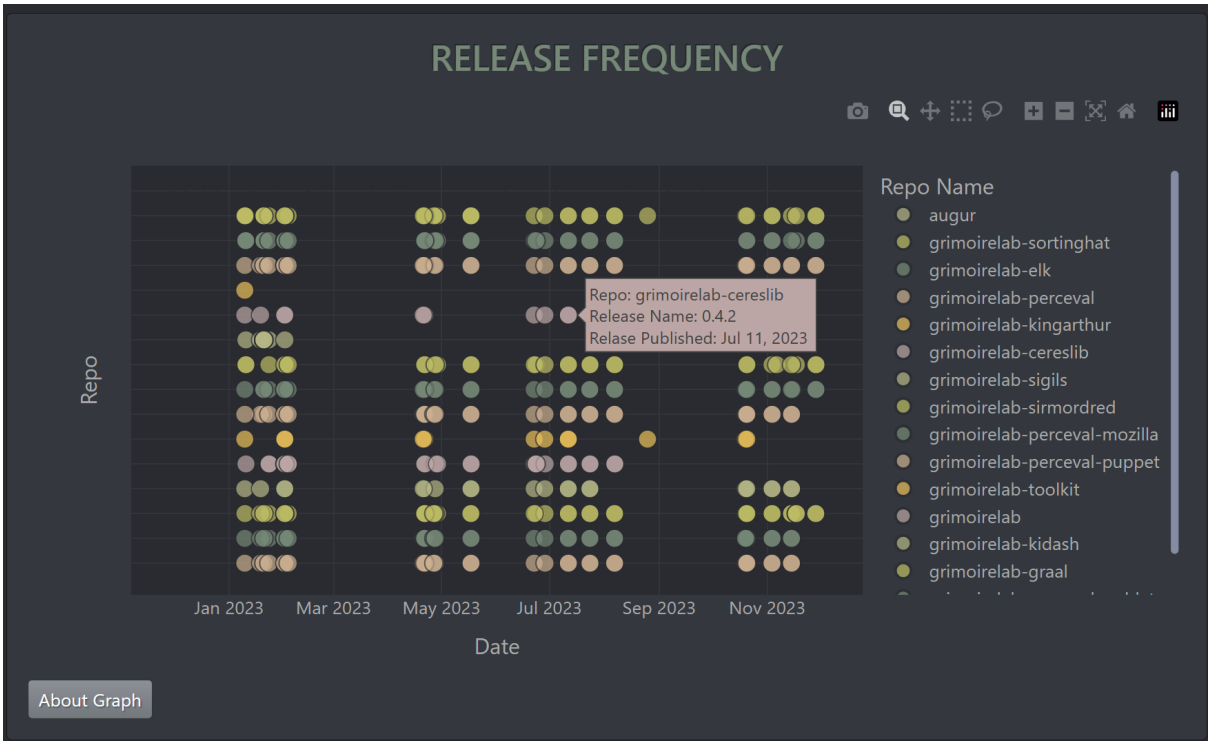
- Contributors
    - Provides a count of the number of contributors, with the parameters of start and end date.
    - You can filter this data by commit authors, issue authors, review participants, mailing list authors, event participants, IRC authors, blog authors, by release cycle, the timeframe of activity, programming languages, or project roles.
- Code Changes Commits
    - Provides a count of the total number of changes during a period, using start/finish dates and source code criteria as parameters.
    - You can filter this data by actors (author, committer), actor demographics, tags (in commit messages), count per month, or count per group over time.
- Activity Dates and Times
    - Provides dates and times of all activity.
    - You can filter this data by individual/organization, aggregation of time by UTC/local time, repository ID, and segment of community.
- Contribution Attribution
    - Provides data about contributors and contributions.
    - Contributors can be filtered by contributor demographics, volunteer and sponsor status, and contributor role. Contributions can be filtered by links to contribution artifacts and indication of contribution types not managed by Git platforms.
- Change Request Reviews
    - Provides the percentage of change requests formally reviewed, the mean/median number of reviews accompanying change requests that are reviewed, and differences in change requests review process that exist among competing open source projects.
    - Data can be filtered by number of unique contributors doing reviews, bot vs humans, change requests accepted/declined/duration/acceptance ratio, time period, and by if the change requests are documented in a CONTRIBUTING.md file.
- Issues Closed
    - Provides a count of issues closed during a period, a ratio of closed issues over a total number of issues during a period, and the number of different kinds of reactions (like a thumbs-up) on issues. The parameters are period of time, source code criteria, criteria for closed, and reopen as new (whether or not reopened issues are considered new issues).
    - Can be filtered by actors or groups of actors.

Starter Project Health contains the following metrics:

- Time to First Response
    - Provides the length of time between an activity and the first time someone responds to it.
    - Can be filtered by role of responder, automated or manual responses, and type of activity.
- Change Request Closure Ratio
    - Provides the ratio between the number of change requests made and the number of change requests closed.
    - Can be filtered by date ranges, automated or manual responses, labels, types of change requests, or types of closures (acceptance or rejection).
- Bus Factor
    - Must be done manually, but the data required is provided. The bus factor measured how many contributors make 50% of the contributions.
    - Can be filtered by time periods or by repository group.
- Release Frequency
    - Provides data about frequency of releases over a specified period of time.
    - Can be filtered by type of release, tag name, count, dates (creation, publication), period of time, and size.

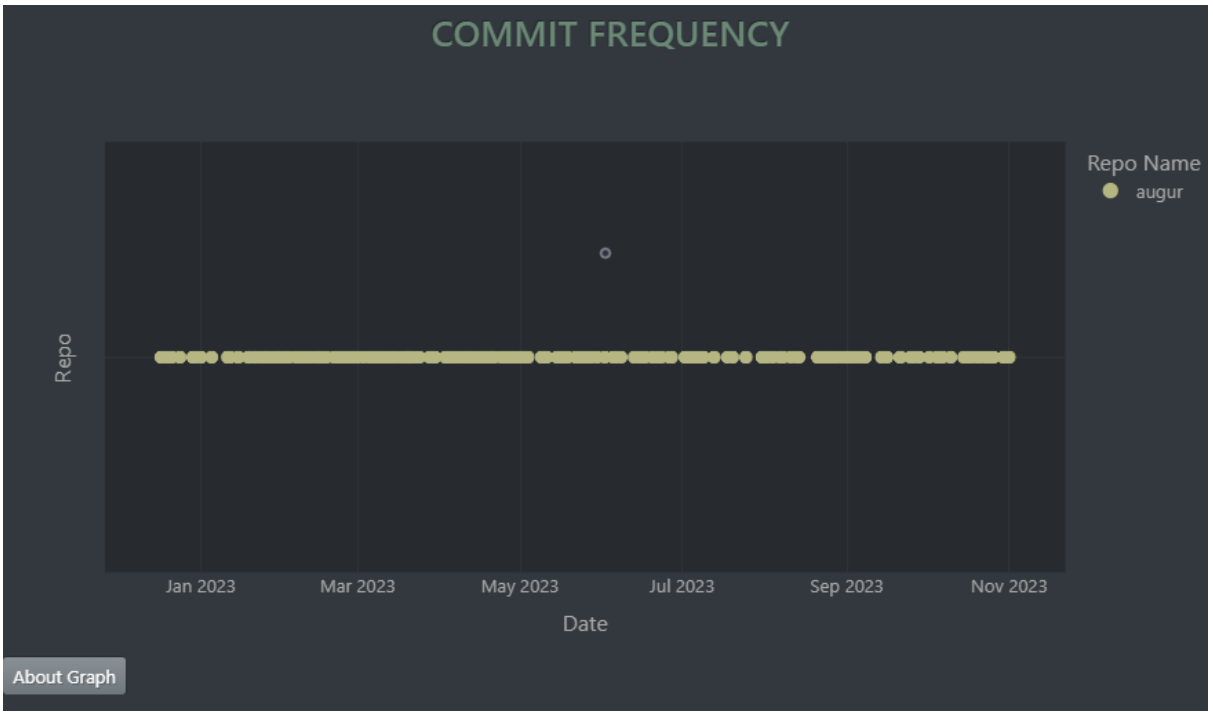
The following are the visualizations implemented in the project using the mentioned metrics.

RELEASE FREQUENCY



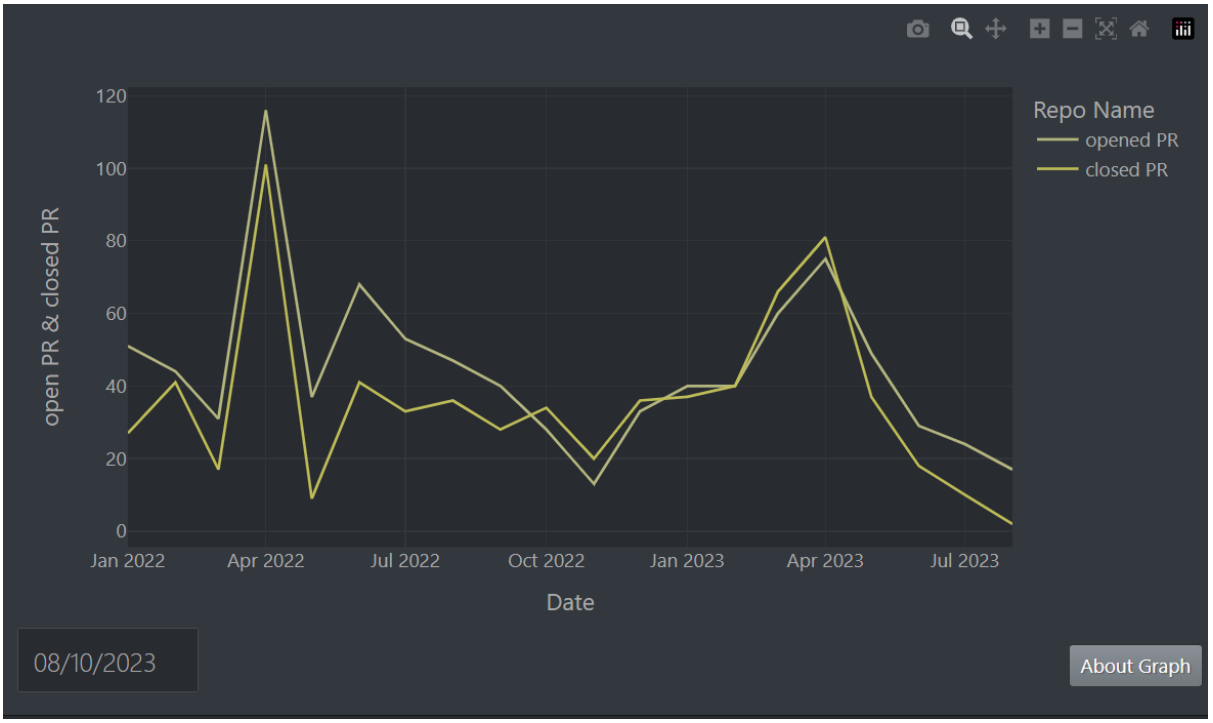
This metric shows the release frequency in the github repo. The visualization differentiates between the different repos by stacking them vertically, and shows each repo update on a horizontal timeline. This should be checked to make sure that the community is staying active. It's easy for any supervisor to see which repos are active, as you can also look at the repo names on the side or by hovering over a data point.

COMMIT FREQUENCY



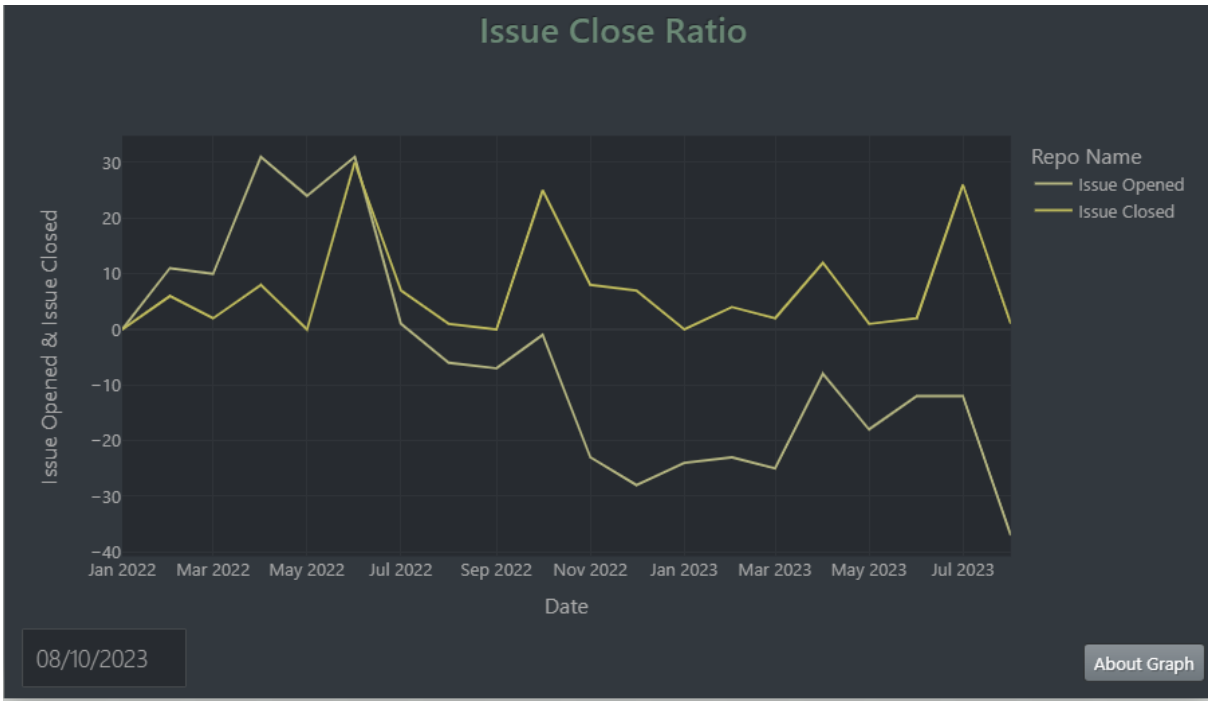
Additionally, the commit frequency visualization shows how often commits are made to the repo, as opposed to releases. Its use is generally the same as the release frequency graph.

CHANGE REQUEST CLOSURE RATIO



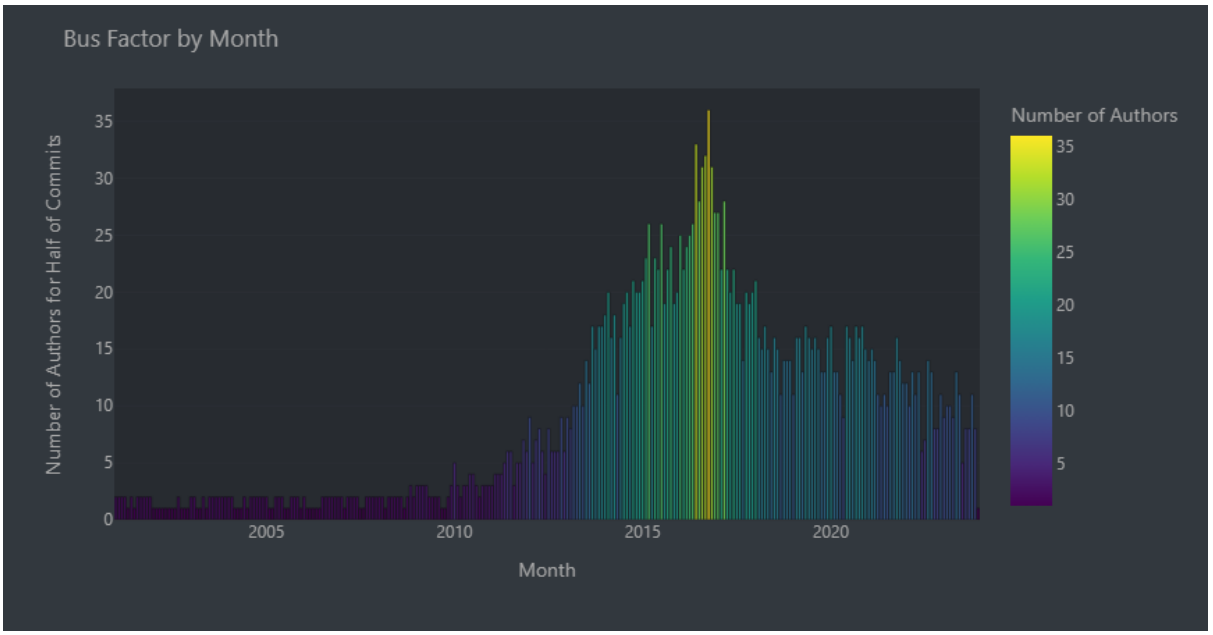
This metric shows the change request closure ratio. The visualization has a range of 2 years before the chosen date, showing how many change requests have been opened and how many have been closed on any given month. The goal of any supervisor should be to keep this ratio relatively even. If the line representing the opened requests is far above the closed line, it means that requests are being unfulfilled. The closed line can appear above the opened line if requests from previous months are closed while fewer are opened in the current month.

ISSUE CLOSE RATIO



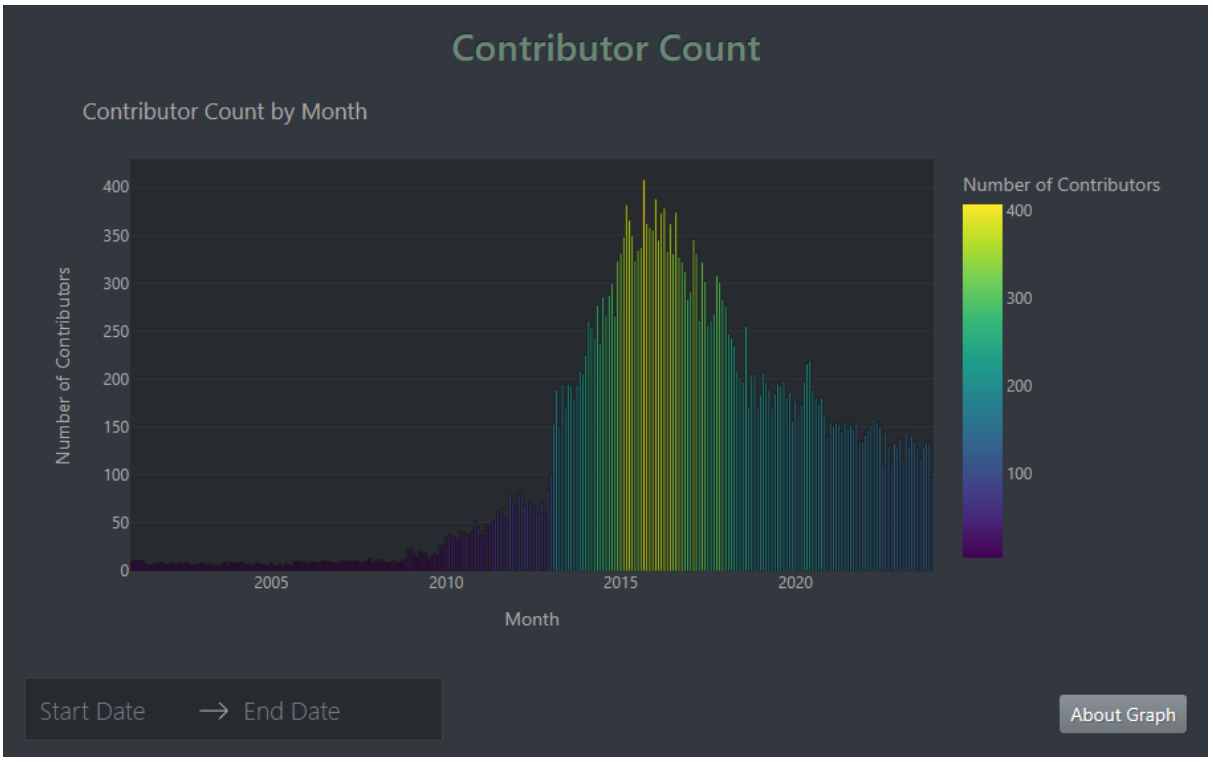
Similarly to the change request close ratio, issues opened and closed are visualized as two lines on a time graph. The goal of any project lead should again be to keep the two lines as vertically close to each other as possible. If issues are occurring and staying unresolved, the ‘issues closed’ line will continue to dip further beneath the ‘issues open’ line.

BUS FACTOR



This metric shows the bus factor. This is used to track the contribution rate of a community. The bus factor tracks the commits to find the lowest number of contributors that add up to half of the commits in a month. If the bus factor is 1, that means that 1 person made half of the commits to the repo. This metric shows the change in bus factor, using months as a time interval. Ideally, a supervisor should be aiming to keep the bus factor as high as possible. A community should not be upheld by a small number of people, because it would mean losing a small number of people would kill an entire project.

CONTRIBUTOR COUNT



This metric shows the amount of contributors active over time. The visualisation shows months on the x axis, and number of contributors on the y axis. Contributors are measured by the amount of members that make any sort of edit or response during the month. This is important to see for any project lead, as it's easy to see how active a community is just by measuring the height of the graph at any given time period. Ideally, the number of contributors should either grow or stay at a steady number.