



HubListener: Software Metrics Analysis & Modelling for Open-Source Projects on Github

Zed Ahmad, Prakhar Jalan, Pedro Oliveira, Piranaven Selvathayabaran
supervised by Dr. Spencer Smith[†]

[†]Department of Computing and Software, McMaster University
1280 Main St. W, Hamilton, Ontario, Canada L8S 4L8

December 5th, 2018



Project Purpose

The HubListener tool was designed to provide users with relevant metrics, trends, and information regarding their GitHub projects. The goal here is to bring this set of organized data to the user’s attention and push them to make any improvements or corrections; as a means of project review and evaluation. Our tool puts a given repository against a set of the most popular projects on GitHub, making comparisons wherever possible. This however begs the question, “What specific metrics lead to success, and how do we measure this success?”

Target Users

The HubListener Tool's general consumer audience would be any member of the open-source community. We have however specifically targetted those with expertise using open-source software. We would hope that individuals within these groups that are responsible for general management of the project would see value in this tool (i.e. Project Managers).

Example User Groups:

- Developers and Creators
- Students (i.e. Engineers)
- Researchers
- Companies/Organizations

As seen in the 'Proof of Concept' section below, a friendly user interface has not yet been implemented, and so the user should at least be familiar with the use of a Command Line Interface.

Proof of Concept

```
{ name: 'HubListener',
  createdAt: '2018-09-27T18:58:42Z',
  description: null,
  forkCount: 0,
  number_of_collaborators: 5,
  number_of_issues: 48,
  number_of_issues_open: 26,
  number_of_issues_closed: 22,
  collaborators:
    [ { login: 'smiths', name: 'Spencer Smith' },
      { login: 'PrakharJalan', name: 'Prakhar Jalan' },
      { login: 'ahmadzed', name: null },
      { login: 'pjmc-oliveira', name: 'Pedro Oliveira' },
      { login: 'selvatp', name: null } ],
  issue_closed_percent: 45.83333333333333,
  issue_closing_times:
    { sum: 482244.7833333333,
      count: 22,
      mean: 21920.217424242423,
      median: 20090.683333333334 },
  issue_open_times:
    { sum: 717821.3807166667,
      count: 26,
      mean: 27608.514642948718,
      median: 21662.333233333333 } }
Successfully cloned repository to C:\Users\Zed\AppData\
{ ': { files: 2, lines: 677 },
  '.aux': { files: 4, lines: 147 },
  '.log': { files: 6, lines: 2586 },
  '.pdf': { files: 4, lines: 6677 },
  '.gz': { files: 4, lines: 540 },
```

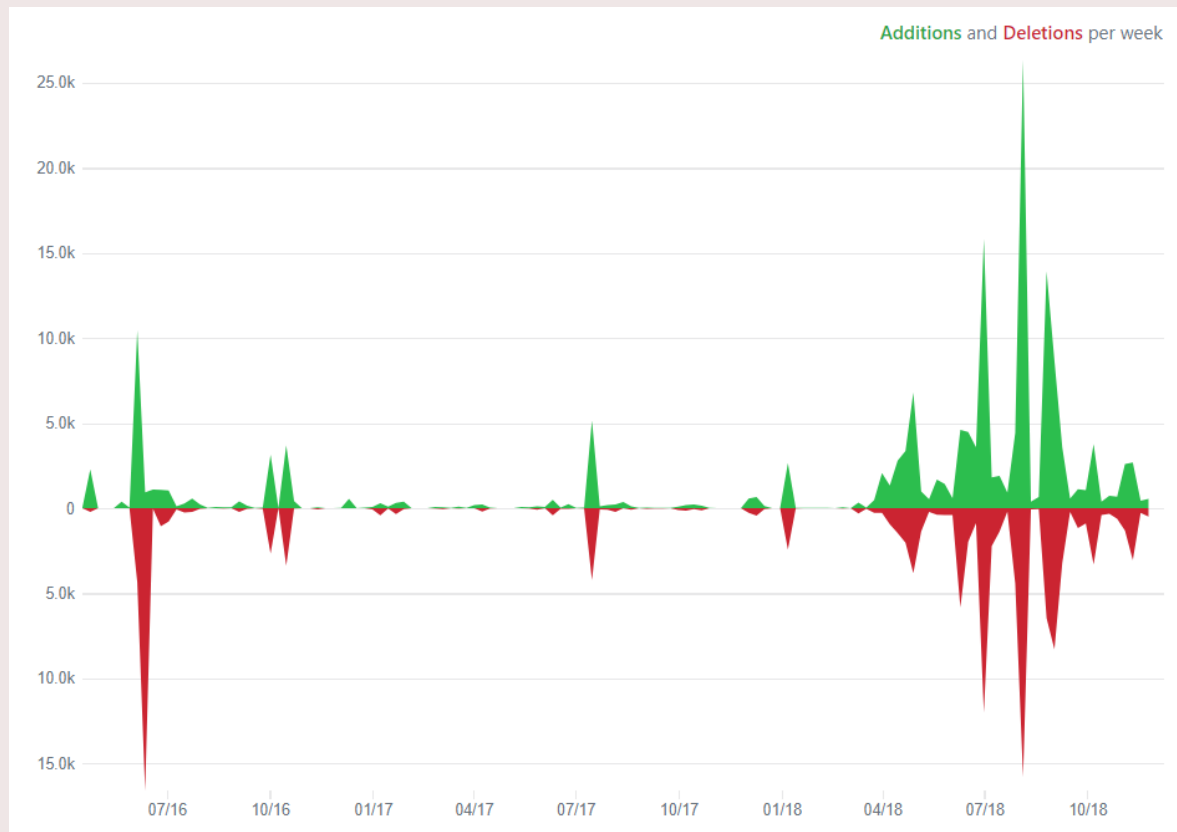
The user enters the project URL via the Command Line Interface, and is provided the output in an array. This output consists of a large list of metrics that are pulled directly from GitHub; which is accomplished through the use of GitHub GraphQL API v4.

Output of Metrics

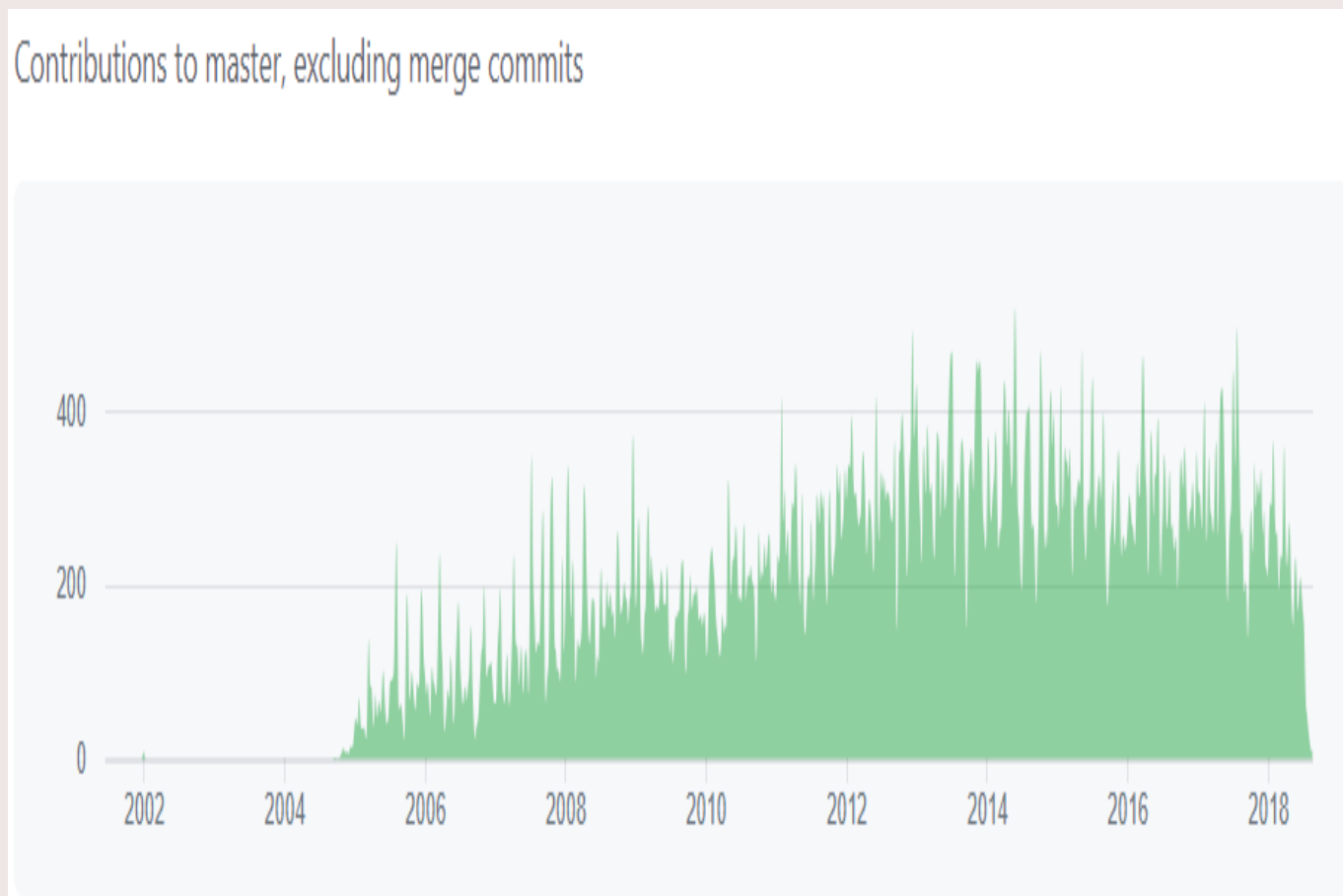
The user provides the GitHub repository link as input, and we return the the output listed below. These metrics are in addition to the existing metrics provided by GitHub.

- Cyclomatic Complexity
- Essential Complexity
- Integration Complexity
- Cyclomatic Density
- Lines of Code
- Lines of Comments
- Maintainability Index
- Coupling Metric
- Number of Methods
- Number of Variables
- Functional Coverage Score
- Condition Coverage Score

Below are some examples of trends that can be viewed via GitHub. Our goal is to provide a trend pairing any of the metrics jotted above (or observe the pattern across time), as GitHub is unable to do so currently.



Example 1: A visualization of additions and deletions from a GitHub repository. This provides the user with a clear understanding of the quantity and frequency of 'change' for the project. A project manager can easily investigate outliers and push for improvements if needed.



Example 2: A visualization of contributions to a repository over time. Enables stakeholders to easily grasp the overall activity of the open-source community on the project. For further analysis, a project manager may compare this data with other similar projects.

Project Goals & Future Work

We hope that our tool will provide valuable metrics or trends to at least 20 users, leading to any changes (minor or major) to their GitHub project; at which point our team can safely say that we have succeeded with this project. Future work includes creating a friendly user interface for easier use of our application, as well as expanding the amount of metrics and trends analyzed. We also wish to add more user options/customizability, so that they are able to be more selective when it comes to the output of HubListener.

Acknowledgements

The HubListener Capstone team would like to thank Dr. Smith for supervising this project. Our shared enthusiasm for this project, along with his consistent input, has been instrumental to our ongoing success. We would also like to thank Dr. Anand for his organization of the course and the Faculty of Engineering at McMaster for providing us the platform to pursue a project of this magnitude. We greatly appreciate all the feedback the students, faculty and staff have provided on this project. Your comments and concerns continue to fuel our passion for HubListener.

References

[1] T. Johnston, “Toolkit for automatic collection and predictive modelling of software metrics,” Master’s thesis, McMaster University, Hamilton, Ontario, Canada, 11 2016. Supervised by Dr. Douglas G. Down.