



Capstone Project Work Report

(Project Term Sept-Apr 2019)

HubListener - GitHub Analysis Tool

Submitted by

**Pedro Oliveira
Zed Ahmad
Piranaven Selva
Prakhar Jalan**

**Student Number: 001430273
Student Number: 1062883
Student Number: 001419766
Student Number: 001450321**

Course Code: 4ZP6B

Under the Supervision of

(Dr. Spencer Smith)

Table of Contents

Table of Contents	1
Introduction	2
Project Purpose	2
Target Users	2
HubListener Tool	2
Scope of Service	2
Static Metrics Analysis	3
Github Metrics Analysis	3
Command-Line Interface (CLI)	4
Web User Interface (WebUI)	4
Future Work	4
Extending the Metrics	4
Extending the Languages	5
Branch Support	5
HubListener Documentation	5
Development Plan	5
Architecture & Design	6
Requirements Specification Document	6
Verification & Validation Plan	6
Poster	7
Presentation & Demo	7
Onboarding	7
Contributing to HubListener	8
Research Material	8
Conclusion	8
Acknowledgments	9

Introduction

This document is meant to provide the reader with a detailed understanding of HubListener and its development history. It will go in depth into the planning, design, testing, and development of the tool as well as providing informal information on the project experience from the perspective of the capstone team. Documentation and code referenced throughout the report are available to be reviewed in its entirety in our public GitHub repository:

<https://github.com/pjmc-oliveira/HubListener>.

Project Purpose

The HubListener tool was designed to provide users with relevant metrics, trends, and information regarding their GitHub projects. The goal was two-fold. Firstly, 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. Secondly, allow researchers to conduct analysis on the open source world in a way that wasn't available before. Our tool gathers statistics by bringing static analysis and Git analysis together to paint a detailed image of a project. Further sorting and querying enhance this image to tell a story. Like any good story, this story has its unanswered questions; just a cliffhanger. "What specific metrics lead to success? How do we measure this success? How do we compare to others?" Given that every project is unique in their own standing, we leave the end user to find the answers to questions like these. Our hope is that HubListener will aid in finding the truth.

Target Users

The HubListener tool's general consumer audience would be any member of the open-source community. We have however specifically targeted those with expertise using open-source software. We would hope that individuals within these groups that are responsible for the general management of the project would see value in this tool (i.e. Project Managers). In saying this, we would also encourage anyone with any coding experience to use our tool if it benefits them!

HubListener Tool

Scope of Service

HubListener aims to provide a service which allows the user to compare his/her GitHub repository or any GitHub open-source repository against similar repositories in the ecosystem.

The end user will be able to attain meaningful information such as metrics or trends that they can use to improve their current project.

Static Metrics Analysis

HubListener currently uses a set of static metrics as part of its static analysis. They are broken down into Raw metrics, Cyclomatic and Halstead metrics. These are listed below:

Raw:

- Number of Files
- Number of Lines
- Number of Logical Lines
- Number of Comments

Cyclomatic:

- Cyclomatic Complexity

Halstead:

- Halstead Effort
- Halstead Bugs
- Halstead Length
- Halstead Difficulty
- Halstead Time
- Halstead Vocabulary
- Halstead Volume

The above metrics are implemented in Javascript as well as Python. They can be extended to any language that is required through our analyse.js module. They were chosen by examining research papers and various open source static analysis tools. The most common statistics were chosen.

Github Metrics Analysis

HubListener currently pulls various GitHub metrics. These are listed below:

Github:

- Number of Open Issues
- Number of Closed Issues
- Total Number of Issues
- Total Pull Requests
- Open Pull Requests
- Closed Pull Requests
- Merged Pull Requests
- Number of Commits in Master
- Number of Forks

Again, these metrics were the most common metrics used on Github from our combined experience. Adding to this list of metrics is very simple given our design architecture

Command-Line Interface (CLI)

HubListener is available as a Command Line Interface. This was implemented before the WebUI and allows HubListener to be used programmatically. This is especially beneficial if you would like to grab data in a different format such as CSV.

The command line interface has the following options:

```
Usage: node app.js [--url <url>] [options...]
```

```
-h, --help          : Print command line options
-u, --url <url>     : GitHub project url
--no-clone          : Don't clone repository
-o, --out <file>    : Optional output file to output results
--csv               : If output file specified, output as CSV
--json              : If output file specified, output as JSON
-a, --append        : Append to file if exists
```

Documentation can be found at:

<https://github.com/pjmc-oliveira/HubListener>

Web User Interface (WebUI)

HubListener is available in the form of a Web User Interface. This was created for the end user to better visualize the metrics and how they compare over time. The chart's x-axis and y-axis contain different metrics while being plotted based on the file extension. The graphs were built with TauCharts.js, a popular open source chart generated. The data can be exported as a JSON file from the web UI via the Export Button. The WebUI version of HubListener can be found on our repository: <https://github.com/pjmc-oliveira/HubListener>

Future Work

The beauty of this project is that it can be extendable in more ways than we can mention. We would love to use the latest technologies such as AI to further enhance our software. Below we have outlined some of the things we would like to do in the future.

Extending the Metrics

One of the major things we can do to enhance our tool is to extend the metrics we provide. This can be done in a variety of ways. Firstly, we can extend the static analysis metrics to include a variety of available and newly researched metrics. Some examples include the following:

- Cyclomatic Density
- Functional Coverage Score
- Condition Coverage Score

Similarly, we would love to add new and improved Github metrics including:

- Commit Comments
- Labels
- Languages

Extending the Languages

HubListener currently only supports two languages, Javascript & Python. Ideally, in the near future, we would love to add support to as many languages as possible. Expansion to different languages opens up further opportunities to gather more statistics. For example, expanding to OOP heavy languages such as Java would allow us to gain metrics such as weighted methods per class and coupling between object class.

Branch Support

The last thing we would like to do to improve HubListener would be to allow for branch support. Currently, the tool only evaluates the master branch. This is useful because the main code should be deployed here. However, we would love to give to the end user the ability to compare their branches such that they can further investigate and analyze their open source project.

HubListener Documentation

Provided below is all of the required documentation for our project, created and completed throughout the year. All of our documentation can be found on the HubListener GitHub repository: <https://github.com/pjmc-oliveira/HubListener/tree/master/Doc>

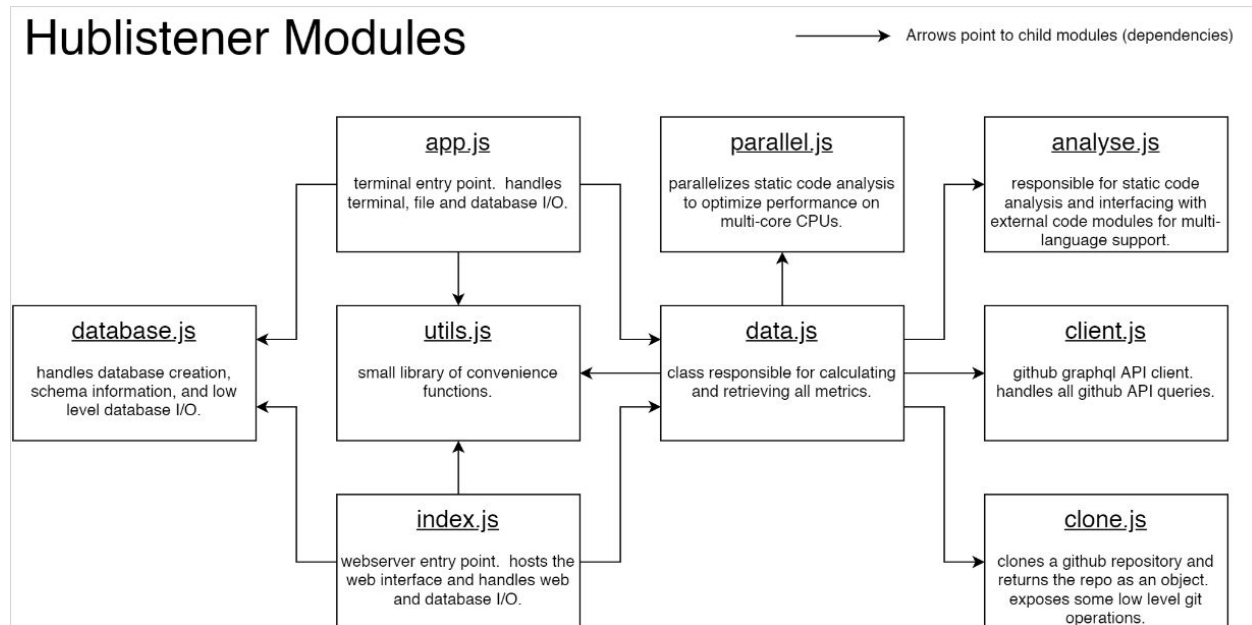
Development Plan

We began this project with an initial development plan that was followed (roughly) to the end. This document outlines our team meeting/communication plan, member roles, technology, coding style, and more:

<https://github.com/pjmc-oliveira/HubListener/blob/master/Doc/DevelopmentPlan/DevelopmentPlan.pdf>

Architecture & Design

Below is a simple visual flow of HubListener's system architecture and module compartmentalization:



Requirements Specification Document

A completed (in-depth) requirements specifications document can be found on the HubListener repository. It outlines the project drivers, project constraints, functional requirements, non-functional requirements, as well as a detailed explanation of our project issues.

Requirements Specification Document (PDF):

<https://github.com/pjmc-oliveira/HubListener/blob/master/Doc/RequirementsSpecificationsDocument/HubListenerRequirementsSpecificationsDocument.pdf>

Verification & Validation Plan

Our verification and validation plan consists of all the relevant verifications needed (SRS, Design, Implementation), along with detailed test cases, and traceability between test cases and requirements.

Verification & Validation Plan (PDF):

<https://github.com/pjmc-oliveira/HubListener/blob/master/Doc/Verification%26ValidationPlan/VnVReport/SystVnVReport/SystVnVReport.pdf>

Poster

Our official capstone poster (created in Latex) was finalized in December of 2018. At the time, HubListener was in its version 1.0 stage of development (Command Line Interface). The poster provided a concise purpose and proof of concept, with appropriate acknowledgments and future considerations account for.

Our experience interacting with students/faculty of McMaster during our open presentation stand that was set up in JHE proved to be a very positive one. With various questions regarding our implementation and overall idea, we learned more about our purpose as a capstone project. We really appreciated all of the positive/critical feedback provided to us.

Poster (PDF): <https://github.com/pjmc-oliveira/HubListener/blob/master/Doc/Poster/Poster.pdf>

Presentation & Demo

HubListener was officially presented and Demoed on April 25th @ ETB 126. The complete powerpoint used is linked below:

<https://github.com/pjmc-oliveira/HubListener/tree/master/Doc/FinalPresentation>

Onboarding

One interesting component of our project was to develop an Onboarding Plan for an incoming member to join our team. The first thing we had to do was develop a job posting that would be used to recruit a new member. For this, we researched job advertisement sites. This led to the creation of the HubListener Job Posting. Following this, we organized all of the documentation, research material, and example code we could, into a 30-day plan which would enable a new member to understand and contribute to our project. This is viewable in our report at Onboarding Plan. Lastly, to enhance our plan we added both an organizational chart and a Survey which would greatly improve our success rate. Overall, it was a great experience to gauge and improve our project organization and quality.

HubListener Organizational Chart, HubListener Posting, Onboarding Plan, and Onboarding Survey (PDF):

<https://github.com/pjmc-oliveira/HubListener/tree/master/Doc/Onboarding>

Contributing to HubListener

HubListener is built in a way that is easy for anyone in the open-source ecosystem to contribute to. We have provided detailed documentation which is in our **Doc/** And **Resource/** folders. On top of this, we created a Github Wiki that further explains how any developer can add to HubListener and the appropriate rules that they should follow while doing so. More details on the HubListener Wiki or contributing to HubListener can be found here:

<https://github.com/pjmc-oliveira/HubListener/wiki>

Research Material

Below are a set of documents that helped with HubListener's direction and purpose throughout its progression over the year.

1. An Empirical Study of Software Metrics
Authors: H. F. LI, Member, IEEE, and W. K. Cheung
2. An Empirical Study on the Relationship among Software Design Quality, Development Effort, and Governance in Open Source Projects
Authors: Eugenio Capra, Chiara Francalanci, and Francesco Merlo
3. Code quality analysis in open source software development
Authors: Ioannis Stamelos, Lefteris Angelis, Apostolos Oikonomou & Georgios L. Bleris
4. Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction
Authors: Tibor Gyimothy, Rudolf Ferenc, and Istvan Siket
5. Software Metrics: Roadmap
Authors: Norman Fenton & Martin Neil

Reference Material: <https://github.com/pjmc-oliveira/HubListener/tree/master/ReferenceMaterial>

Conclusion

Working on the HubListener project was a very fulfilling experience. Being able to come up with an idea and then be a part of all the steps necessary to turn that idea into a useable software was something truly rewarding. Throughout the project, we faced many challenges (both professional and personal) and we are proud to say that we have overcome all of them. As a group, we learned many things about project management, documentation, and development. We are pleased and excited to translate the knowledge we gained and the skills developed into our future endeavors. We are very proud of our year-long effort and hope to work on it again in the future or watch it grow in the open source world!

Acknowledgments

There are many people we would like to thank for their contribution to HubListener's success. Most notably, Dr. Smith, for supervising this project. Our shared enthusiasm for this project, along with his consistent and meaningful input, was instrumental to our success. It was an absolute pleasure working with him and we were humbled to have had him as a supervisor.

We would also like to take this opportunity 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.