GitHub Online Analytics

Moritz Fuller[a]

[a] *Humboldt Universitaet zu Berlin, Unter den Linden 6, Berlin, 10117, Berlin, Germany*

Abstract

GOA (GitHub Online Analytics) is a web application that provides a graphical overview of the evolution of a GitHub repository.

1. Motivation of research problem and research question

There are many tools to analyze GitHub repository data, but from a first shallow research each of them comes with certain drawbacks. You either need to install something, configure a complicated setup, pay for it, the use case is too specific or it's focussing more on data around the git protocol itself rather than the social part of GitHub . What's missing is a lightweight, flexible and easy to use GitHub Online Analytics (GOA) tool.

The question we are trying to answer in our thesis is what kind of tools are out there, how do they function and what kind of features do they provide. We want to achieve this by developing a framework to compare each one of the services using certain criteria that are being explained in more detail in section 3. Using this comparison we can identify a niche, an ideal product that does not exist yet and that we are going to develop then.

The next step would be to find out what kind of data should be visualized and how it should be visualized. We will try to use the ESeVis framework developed by Yeshchenko and Mendling (2022) to categorize the visualization techniques used by current solutions, although it seems to focus on event sequence data, which is not our primary focus for this tool. At the same time we want develop a framework that categorizes the different data points that are available for analysis. We might settle on the ”Common Metrics” defined by CHAOSS (b).

For now the plan is to develop a webapp where you can enter the link to a GitHub repository that should be analyzed. The webapp will then make calls to the GitHub GraphQL API to retrieve the data and visualize it using a framework like d3.js or ECharts. Ideally the user should be able to configure what data is being analyzed and how it should be visualized using a simple interface.

2. Summary of background literature and state of the art solutions

There are numerous tools out there that can be used to analyze GitHub repositories. In the following chapters we will give a quick overview of the ones that are closest to what we are trying to achieve and outline their characteristics.

2.1. Apache Kibble

Apache Kibble is a suite of tools for collecting, aggregating and visualizing activity in software projects developed by the Apache Software Foundation. It has to be deployed manually and can be configured to scan multiple different data sources like GitHub, JIRA or a mailing list. It has no releases, but users can try the development version. The project seems abandoned as the last commit to main as of writing this was 13 months ago. The configuration and setup are non trivial and require a good knowledge of the underlying technologies like UNIX systems, Python and Apache HTTP Server (The Apache Software Foundation).

2.2. Augur

Augur is a Python library and web service for Open Source Software Health and Sustainability metrics & data collection. It has to be deployed manually and can be configured to scan multiple different data sources like GitHub, git commit logs and the Core Infrastructure Initiatives API. It is actively maintained and while there a Docker images available for a quick deployment, when used for long-term data collection it still requires non-trivial installation steps. Those include setting up a PostgreSQL database, installing the instance, an application server and Augur's data collection workers (CHAOSS, a).

2.3. Gitinspector

Gitinspector is a statistical analysis tool for git repositories. It can be installed locally using `npm` or a package manager like `apt-get` and provides a command line utility. `Python` and `git` need to be present on the system, together with the repository that should be analyzed. It focuses on a single repository and enables users to compare distinct contributors. It is mainly used as a grading aid by universities. Although the project has 2.1k stars on GitHub, it seems abandoned, as the last commit was 2 years ago (Ejwa Software).

2.4. Frontend Repo Analyzer

The Frontend Repo Analyzer is a tool that can analyze `git` repositories and report metrics about them. It supports multiple repositories and can be configured for a variety of metrics. As the name suggests it focuses more on repositories that contain Frontend code, it does read the dependencies from a `package.json` file. or the CSS bundle size for example. The installation is quite easy, as it's available for `npm`, but there is still some setup left to do for the data base which stores the results after each run and the configuration of the tool. The project seems to be maintained (Feedzai).

2.5. Grimoire Lab

Grimoire Lab provides a coordinated set of tools to retrieve data from systems used to support software development (repositories), store it in databases, enrich it by computing relevant metrics and making it easy to run analytics and visualizations on it. Although they provide a docker image to ease start playing, if used in a production environment extensive configuration and setup of each one of the tools is required. For an overview see Figure 1. It is actively maintained and there are multiple projects building on this stack, like Cauldron 2.8 and Insights 2.7 (Dueñas et al., 2021; CHAOSS, c).

2.6. Haystack

Haystack is a tool to provide insights from Git data, to help experiment faster, ship reliably and prevent burnout. So far it is the tool that seems to come the closest to what we are trying to achieve apart from the difference that it's closed source and a paid service. Users only need to create an Haystack account, connect their version control system and choose which repositories to plug into Haystack. It's not entirely clear what the dashboard

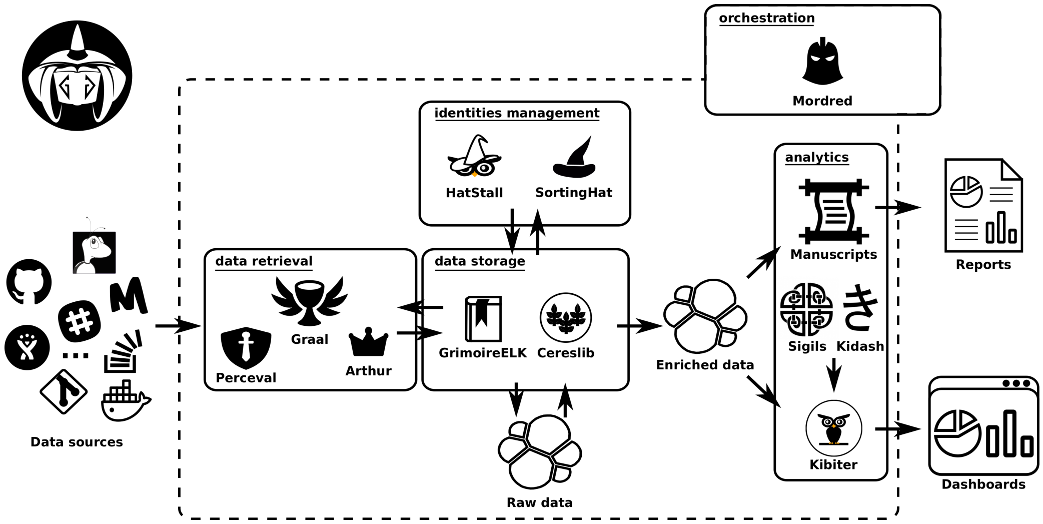

Figure 1: Grimoire Lab tool stack (CHAOSS, c)

provides, as there is no accessible demo available (Haystack - Analytics for Engineering Leaders).

2.7. LFX Insights

LFX Insights provides data-driven insights to make informed decisions. It follows the best practices and specifications from the CHAOSS project, with some of the components being initially sourced from Grimoire Labs. Users are able to connect multiple data sources like emails, GitHub, Google Groups or Twitter to a project. There is no obvious self hosting available and enrolling on their instance your own project requires you to create an account to submit a ticket for onboarding (Insights).

2.8. Cauldron

Cauldron is another project that builds on the Grimoire Lab stack. It is a web application that allows users to create so called reports that contain a set of repositories from multiple data sources. Cauldron then collects and analyzes the data to display it in a dashboard. Reports created with Cauldron are persisted and visible for everyone. It's possible to setup your own paid instance of Cauldron via Cauldron Cloud (Cauldron Cloud) or to deploy the project yourself. The dashboard contains a variety of graphs and analytics out of the box and can be further customized and extended using the Elastic dashboard tied to every project. Apart from that it's possible to compare different projects. Cauldron seems to be maintained, but rather irregular. It's another tool that is very close to what we are trying to achieve. (Cauldron / cauldron; Level up Software Development Analytics - Cauldron).

2.9. Monocle

Monocle enables users to organize daily duties and to detect anomalies in the way changes are produced and reviewed. They provide a docker deployment got get started quickly, but they also document deploying from source. When deploying a `config.yaml` file has to be present, which – among other things – specifies the repositories to be analyzed. It comes with a sensible set of default visualizations that are configurable. Monocle is actively developed (change-metrics).

2.10. RepoSense

RepoSense is a tool to visualize programmer activities across git repositories. You can either use it locally – which requires `Java` and `git` – or remotely using a service like `Netlify` or `GitHub Actions` to generate so called reports. A report consists of an interactive website and can be viewed using a browser. The reports can be customized to a degree using CLI flags and config files, but visualization is rather rigid. The project is actively maintained (RepoSense).

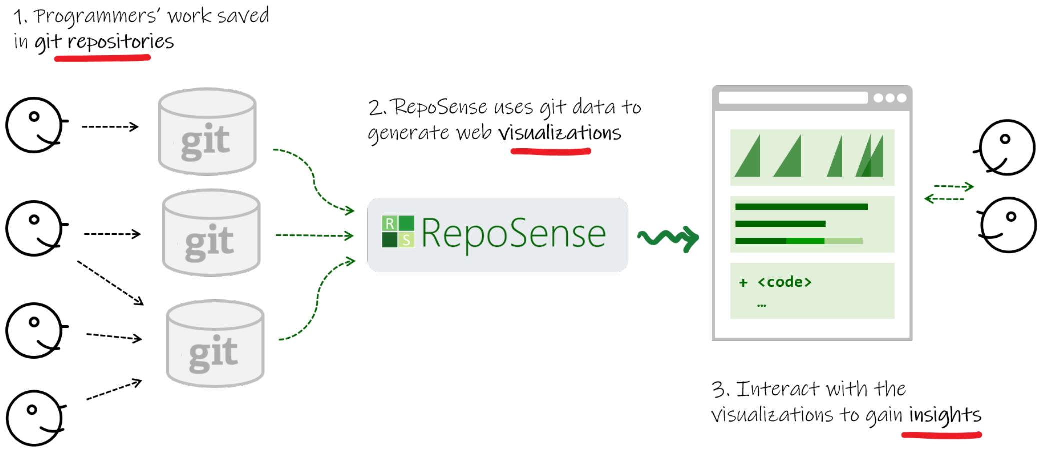
1. Programmers' work saved
in git repositories
git
git
git
2. RepoSense uses git data to
generate web visualizations
RepoSense
+ <code>
…
3. Interact with the
visualizations to gain insights

Figure 2: RepoSense overview (RepoSense - Home)

3. Proposed research method

First we want to analyze the current landscape of available tools and compare them in terms of functionality, events processed and visualization techniques.

3.1. Functionality

The following properties are used to compare the functionality of the tools:

3.1.1. hosted

This property describes wether the tool is hosted or requires deployment.

3.1.2. maintained

This property describes wether a tool is actively maintained or not. To be considered actively maintained there have to be at least 5 commits to the release branch within the past 6 months.

3.1.3. difficulty

The difficulty is described using the following categorization:

- easy : no installation, no setup, plug & play
- medium : requires some setup, but easy to use (e.g download a package or run a docker image)
- hard: requires setting up databases, servers, configurations and use of the CLI

3.1.4. comparisons

This characteristic describes wether the tool allows to compare at least two different repositories with one another.

3.1.5. customizable / explorable

We evaluate wether the tool is customizable in some way. This means that the user can configure different metrics and visualizations with low effort and visually explore the data.

3.1.6. open source

We evaluate wether or not the projects source code is open source to allow community contributions and further enhancements.

3.1.7. repository focused

We evaluate wether the tool is focused on single repositories or on general data.

3.1.8. free

We evaluate wether the tool is free or requires a paid license.

3.1.9. database

Does the tool make us of it's own database to store the data or rely on a third party database.

3.2. Metrics

We will most likely use the "Common Metrics" provided by CHAOSS (b) to compare the different tools out there. The working group focuses on defining metrics that are used by multiple working groups or are important for community health. The focus areas of those metrics are:

- contributions
- people
- place
- time

3.3. Visualization

We could analyze existing solutions regarding the visualizations they are using, although we are not sure yet how useful this would be for our goals. Nevertheless the following properties could be used:

- scatter plot
- theme river
- bar chart
- area chart
- line chart
- heatmap
- stacked bar chart
- pie chart
- bubble chart
- tree map
- radar chart
- ESeViz charts (Yeshchenko and Mendling, 2022)

4. Outline of thesis

1. Introduction
    - 1.1 Background
    - 1.2 Terminology
    - 1.3 Research Question
    - 1.4 Structure
2. Evaluating state of the art solutions

    2.1 Functionality

    2.2 Metrics

    2.3 (Visualization)

    2.4 The ”perfect” tool

3. Technologies

    3.1 Svelte

    3.2 Tailwind

    3.3 d3 — ECharts — Vega-Lite

    3.4 Apollo Svelte

4. Artifact

5. Conclusion

5. Preliminary literature list

See 6 for the complete list.

6. Work plan including milestones

due date	task
15.05.22	finish evaluation
18.05.22	identify niche and specify product
25.05.22	specify technologies
01.07.22	produce artifact
01.08.22	submit thesis

Table 1: Work plan

List of Figures

List of Tables

References

van der Aalst, W.M.P., 2022. How to Write Beautiful Process-and-Data-Science Papers? arXiv:2203.09286 [cs] URL: http://arxiv.org/abs/2203.09286, arXiv:2203.09286.

Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C., 2008. Visual Methods for Analyzing Time-Oriented Data. IEEE Transactions on Visualization and Computer Graphics 14, 47–60. doi:10.1109/TVCG.2007.70415.

Analyzing popular repositories on GitHub, 2021. Analyzing popular repositories on GitHub. URL: https://www.analyticsvidhya.com/blog/2021/07/analyzing-popular-repositories-on-github/.

Anderl, T., 2021. Identifying GitHub Trends Using Temporal Analysis. Thesis. doi:10.34726/hss.2021.93182.

Apache ECharts, 2022. Apache ECharts. URL: https://echarts.apache.org/en/index.html.

Bostock, M., 2022. D3.js - Data-Driven Documents. URL: https://d3js.org/.

Cabot, J., 2021. 20+ tools to help you mine and analyze GitHub and Git data. URL: https://livablesoftware.com/tools-mine-analyze-github-git-software-data/.

Cánovas, J., 2017. All we have learned about software development by mining GitHub (plus some concerns). URL: https://livablesoftware.com/learned-software-development-mining-github-plus-concerns/.

Cánovas Izquierdo, J.L., Cosentino, V., Rolandi, B., Bergel, A., Cabot, J., 2015. GiLA: GitHub label analyzer, in: 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER), pp. 479–483. doi:10.1109/SANER.2015.7081860.

Cauldron / cauldron, 2022. Cauldron / cauldron. URL: https://gitlab.com/cauldronio/cauldron.

Cauldron Cloud, 2022. Cauldron Cloud. URL: https://cloud.cauldron.io/.

change-metrics, 2022. Monocle. URL: https://github.com/change-metrics/monocle.

CHAOSS, 2022a. Augur. URL: https://github.com/chaoss/augur.

CHAOSS, 2022b. CHAOSS Common Metrics Working Group. URL: https://github.com/chaoss/wg-common.

CHAOSS, 2022c. GrimoireLab. URL: https://github.com/chaoss/grimoirelab.

Chart.js — Open source HTML5 Charts for your website, 2022. Chart.js — Open source HTML5 Charts for your website. URL: https://www.chartjs.org/.

Chatziasimidis, F., Stamelos, I., 2015. Data collection and analysis of GitHub repositories and users, in: 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1–6. doi:10.1109/IISA.2015.7388026.

Dueñas, S., Cosentino, V., Gonzalez-Barahona, J.M., Felix, A.d.C.S., Izquierdo-Cortazar, D., Cañas-Díaz, L., García-Plaza, A.P., 2021. GrimoireLab: A toolset for software development analytics. PeerJ Comput. Sci. 7, e601. doi:10.7717/peerj-cs.601.

Ejwa Software, 2022. Ejwa/gitinspector. URL: https://github.com/ejwa/gitinspector.

Feedzai, 2022. Frontend Repo Analyzer. URL: https://github.com/feedzai/repo-analyzer.

GitHub Repo Analysis, 2022. GitHub Repo Analysis. URL: https://kaggle.com/tsalitzion/github-repo-analysis.

Guo, Y., Guo, S., Jin, Z., Kaul, S., Gotz, D., Cao, N., 2021. A Survey on Visual Analysis of Event Sequence Data. IEEE Transactions on Visualization and Computer Graphics , 1–1doi:10.1109/TVCG.2021.3100413.

Haystack - Analytics for Engineering Leaders, 2022. Haystack - Analytics for Engineering Leaders. URL: https://www.usehaystack.io/?utm_campaign=Use-Cases%20Campaign&utm_source=google&utm_medium=cpc&utm_content=GitHub%20Dashboard&utm_term=git%20repository%20statistics&gclid=EAIaIQobChMI4efY64OC9wIViIbVCh2aIgXuEAMYASAAEgICO_D_BwE.

Insights, 2022. Insights. URL: https://insights.lfx.linuxfoundation.org/projects/korg/dashboard;quicktime=time_filter_3Y.

Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P., 2012. Process diagnostics using trace alignment: Opportunities, issues, and challenges. Information Systems 37, 117–141. doi:10.1016/j.is.2011.08.003.

Kumar J., M., Dubey, S., Balaji, B., Rao, D., Rao, D., 2018. Data Visualization on GitHub Repository Parameters Using Elastic Search and Kibana, in: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 554–558. doi:10.1109/ICOEI.2018.8553755.

Level up Software Development Analytics - Cauldron, 2022. Level up Software Development Analytics - Cauldron. URL: https://cauldron.io/.

Li, D., Mei, H., Shen, Y., Su, S., Zhang, W., Wang, J., Zu, M., Chen, W., 2018. ECharts: A declarative framework for rapid construction of web-based visualization. Visual Informatics 2, 136–146. doi:10.1016/j.visinf.2018.04.011.

Liao, Z., He, D., Chen, Z., Fan, X., Zhang, Y., Liu, S., 2018. Exploring the Characteristics of Issue-Related Behaviors in GitHub Using Visualization

Techniques. IEEE Access 6, 24003–24015. doi:10.1109/ACCESS.2018.2810295.

Ludwig, J., Xu, S., Webber, F., 2017. Compiling static software metrics for reliability and maintainability from GitHub repositories, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 5–9. doi:10.1109/SMC.2017.8122569.

RepoSense, 2022. RepoSense. URL: https://github.com/reposense/RepoSense.

RepoSense - Home, 2022. RepoSense - Home. URL: https://reposense.org/index.html.

Rusk, D., Coady, Y., 2014. Location-Based Analysis of Developers and Technologies on GitHub, in: 2014 28th International Conference on Advanced Information Networking and Applications Workshops, pp. 681–685. doi:10.1109/WAINA.2014.110.

Schumann, H., Müller, W., 2000. Visualisierung. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-57193-0.

Shneiderman, B., 2003. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, in: Bederson, B.B., Shneiderman, B. (Eds.), The Craft of Information Visualization. Morgan Kaufmann, San Francisco. Interactive Technologies, pp. 364–371. doi:10.1016/B978-155860915-0/50046-9.

Sundar, D.S., Kankanala, M., 2015. Analyzing and predicting Lifetime of trends using social networks, in: 2015 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–7. doi:10.1109/ICCCI.2015.7218090.

The Apache Software Foundation, 2022. Apache Kibble. URL: https://github.com/apache/kibble.

Vega, 2022a. Vega: A Visualization Grammar. URL: https://github.com/vega/vega.

Vega, 2022b. Vega-Lite. URL: https://github.com/vega/vega-lite.

Weicheng, Y., Beijun, S., Ben, X., 2013. Mining GitHub: Why Commit Stops – Exploring the Relationship between Developer's Commit Pattern and File Version Evolution, in: 2013 20th Asia-Pacific Software Engineering Conference (APSEC), pp. 165–169. doi:10.1109/APSEC.2013.133.

Yeshchenko, A., Mendling, J., 2022. A Survey of Approaches for Event Sequence Analysis and Visualization using the ESeVis Framework. arXiv:2202.07941 [cs] URL: http://arxiv.org/abs/2202.07941, arXiv:2202.07941.