

Visualization of Open Source Projects

Analyzing trends in Github Organizations

David Leonard

DrkSephy1025@gmail.com

Liron Shimrony

lsxliron@gmail.com

Date: May 1, 2016

Contents

1	Introduction	2
2	Why?	2
2.1	Code Swarm	2
2.2	Gource	3
2.3	Github Visualizer	4
3	What?	4
3.1	Event Bar Chart	5
3.2	Event Time Series	5
3.3	Repository Treemap	5
3.4	Interactivity	5
4	How?	5
5	So What?	5
6	Conclusion	5

1 Introduction

One of the biggest and most interesting data sources online is **source code** stored in various version control platforms, such as **Github** and **Bitbucket**. These platforms contain various projects stored in **repositories**, which contain various metadata with which we can draw several conclusions from. This metadata consists of metrics such as:

- **Commits:** Snapshots of the codebase at a given point in time.
- **Issues:** Feature requests and bug reports for a given project.
- **Pull Requests:** Code patches submitted by various other contributors around the world.

The challenge here is to discover various trends in the data with which may not be visible depending on the visualizations chosen to represent the data. However, it is infeasible to analyze every single repository across Github or Bitbucket for this project - as the number of repositories is in the millions. In order to draw meaningful conclusions from the data, we will be selectively analyzing open source projects on Github pertaining to the **Facebook** organization. The choice for this is arbitrary, but we chose Facebook because it is responsible for open sourcing various projects in the last few years which have heavily influenced the open source community. From this data, we expect to draw various conclusions regarding the lifespan of their open source projects and how they progress over time.

2 Why?

Given the general form of the data, the big challenge is to come up with a meaningful visualization which allows us to derive some conclusions. There have been several projects which strictly visualize **commits** in a given repository, which typically generate a tree-like structure (or a structure in general). These visualizations show how large a project gets over the course of time as the number of developers and commits increases. While the structures that are generated at the end of these visualizations are interesting to look at, it is difficult to draw any sort of meaningful conclusions. In the next few sections, we briefly show some popular works in this domain.

2.1 Code Swarm

Code Swarm [2] is a visualization of commits in a repository, which was also created by Michael Ogawa [1]. Code Swarm represents files as moving elements, where both files are mapped to the developer which created it. Moreover, these files are color coded based on whether or not they are source code or general documents. When developers drop off in their contributions, they fade over time from the underlying histogram at the bottom which allows us to keep track of what events have occurred before in the timeline. Code Swarm is able to visualize a wide spectrum of repository types:

- Subversion
- CVS
- Git
- Mercurial
- Perforce
- VSS
- Starteam
- Wikiswarm
- Darcs



Figure 1: Code Swarm

2.2 Gource

Another popular source code visualization tool is **Gource**, which also generates a tree-like structure of the repository similar to Code Swarm. Gource is able to visualize **git**, **subversion**, **mercurial** and **bazaar** repositories. These visualizations consist of the root directory being mapped to the root of the tree, while directories will become branches and subsequent files become leaf nodes as time elapses from the conception of the repository to the last commit.

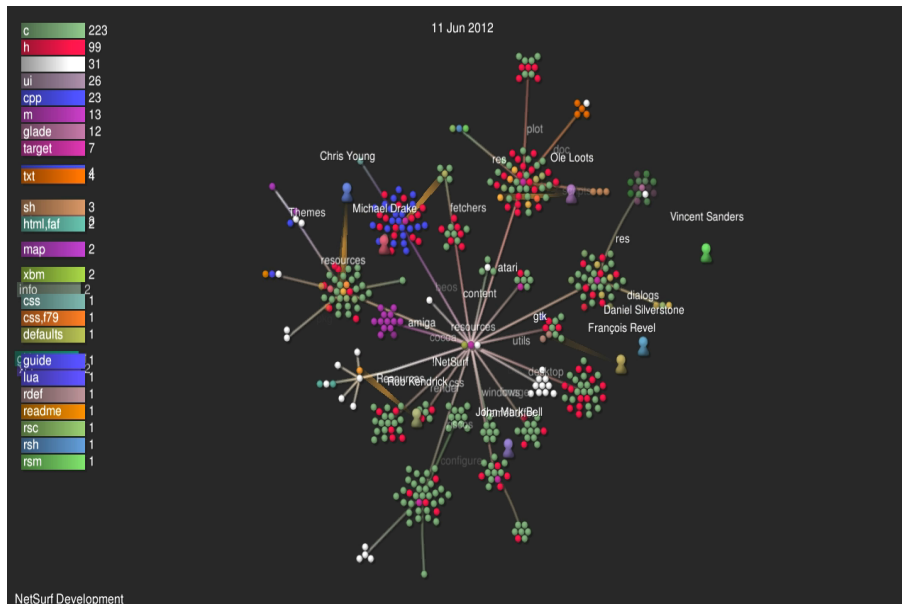


Figure 2: Gource Visualization

2.3 Github Visualizer

One of the most comprehensive open source visualization tools is **Github Visualizer**[4], which was built using d3¹ and the Github API². Using this tool, one may visualize events in a Github repository ranging from commits to lines of code over time. During the runtime of the visualization, a histogram is built along the time-axis showing the density of events for the current time frame. Moreover, a user may choose to visualize an individual repository belonging to a user, or they may visualize all repositories which belong to a given user. By hovering over various elements in the visualization, we can drill down deeper and analyze the lines of code that are added over time per time series event. For reference, we have included a visualization of Facebook repositories generated by this tool in Figure 3.

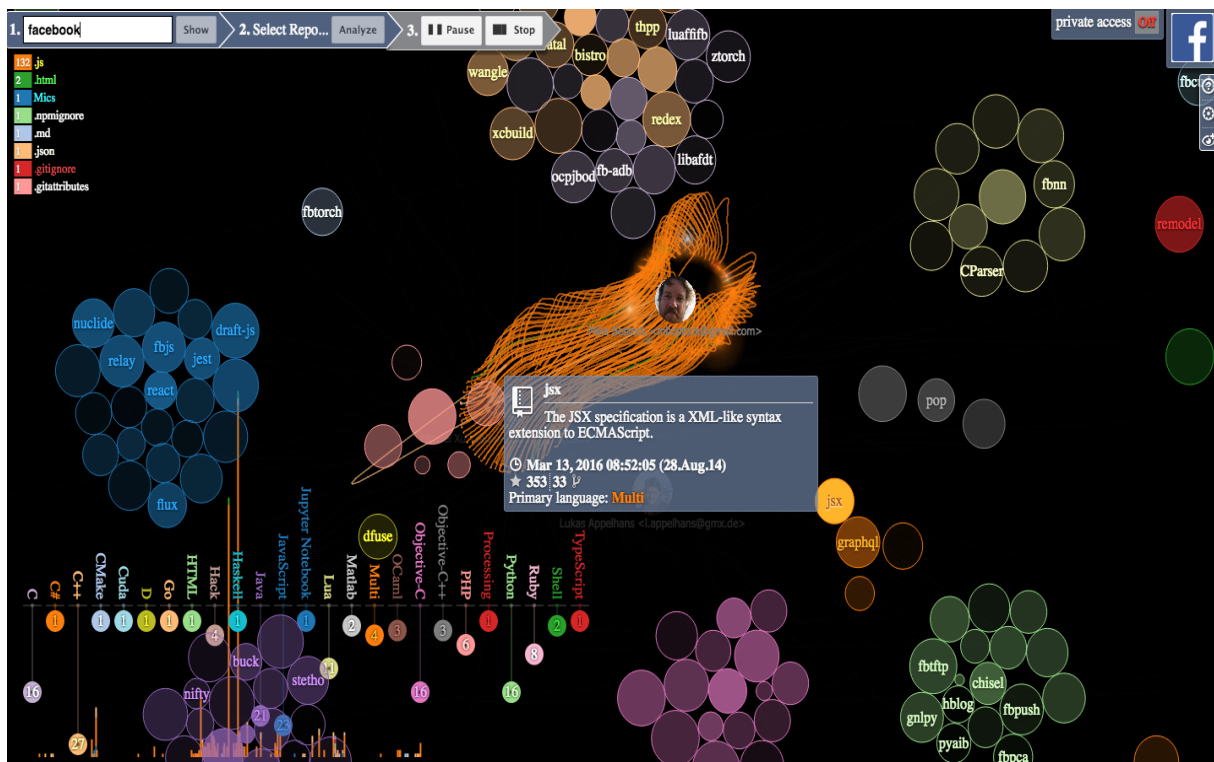


Figure 3: Github Visualizer: Facebook Repositories

While this visualization is very interesting to look at, we felt that it is too dense and it becomes a challenge to narrow down on a particular aspect of the visualization. It is hard to draw any solid conclusions from what is presented, and trends are hard to discover apart from the event density histogram. In our own project, we aim to build a series of visualizations which are simple yet powerful with respect to the trends that can be discovered.

3 What?

In this section we discuss the visualizations which we have created over the span of this project. As mentioned earlier, this project was created without knowing the questions we want to answer, but we expect several noticeable trends to crop up throughout the lifespan of these open source projects. We begin by discussing our first visualization: the **event bar chart**.

¹d3: <https://d3js.org/>

²Github API: <https://developer.github.com/v3/>

3.1 Event Bar Chart

Our first goal was to show the sum totals of various events

3.2 Event Time Series

3.3 Repository Treemap

3.4 Interactivity

4 How?

5 So What?

6 Conclusion

References

- [1] Michael Ogawa, and Kwan-Liu Ma *Software evolution storylines*. Proceedings of the 5th international symposium on Software visualization. ACM, 2010.
- [2] Michael Ogawa. *Code Swarm*. <http://vis.cs.ucdavis.edu/~ogawa/codeswarm/>
- [3] Andrew Caudwell. *Gource*. <https://github.com/acaudwell/Gource>
- [4] Artem Zubkov. *Github Visualizer*. <http://ghv.artzub.com/>