



**I.T.A.K.E.**  
Unconference

**2016**

# Visualizing code bases

@codingandrey

01

# About me

02



# Andrey Adamovich

- Java/Groovy developer, clean coder
- DevOps guy, automation junkie
- Co-author of Groovy 2 Cookbook
- Co-organizer of @latcraft and @devternity
- Coach at @devchampions
- Twitter: @codingandrey

# Let's start!



# Background

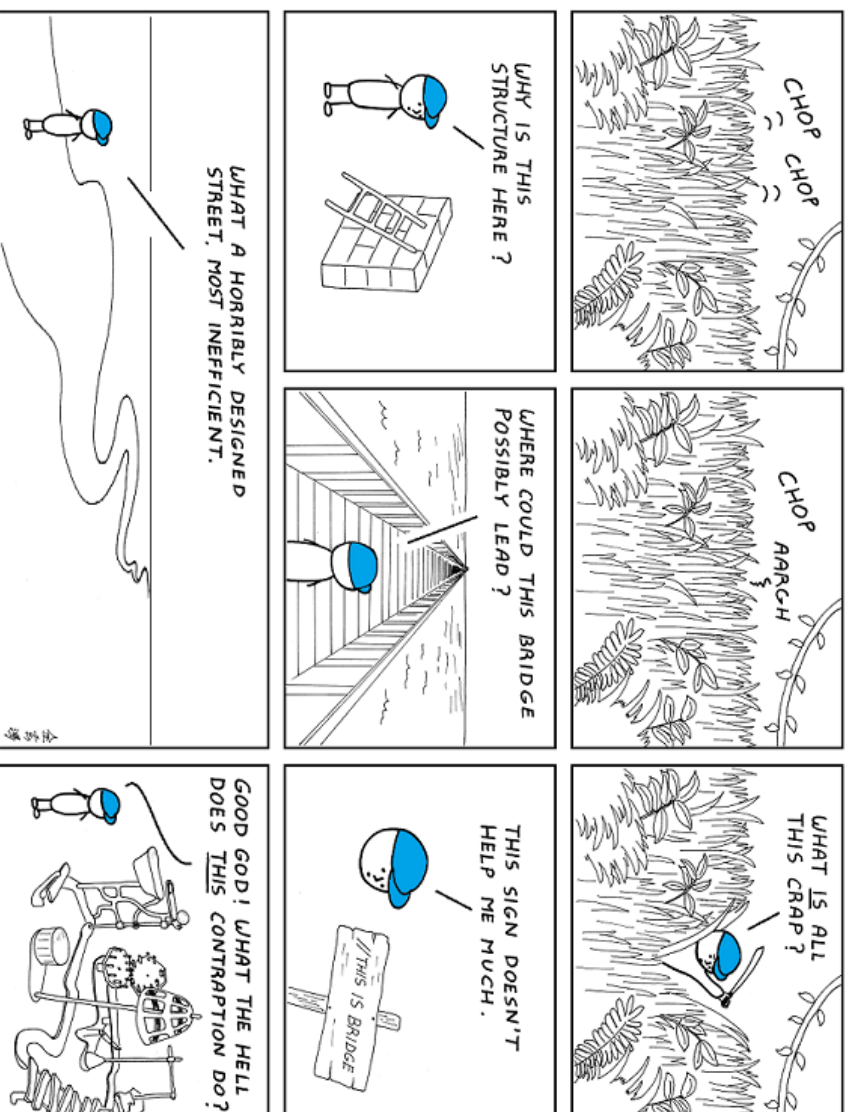
- Find new clients
- Face big code bases
- Need quick analysis
- Need quick results



# Code size

- Google: 2 billions LOC
- Facebook: 61 million LOC
- Me: from 20K to 2M LOC

# I hate reading...

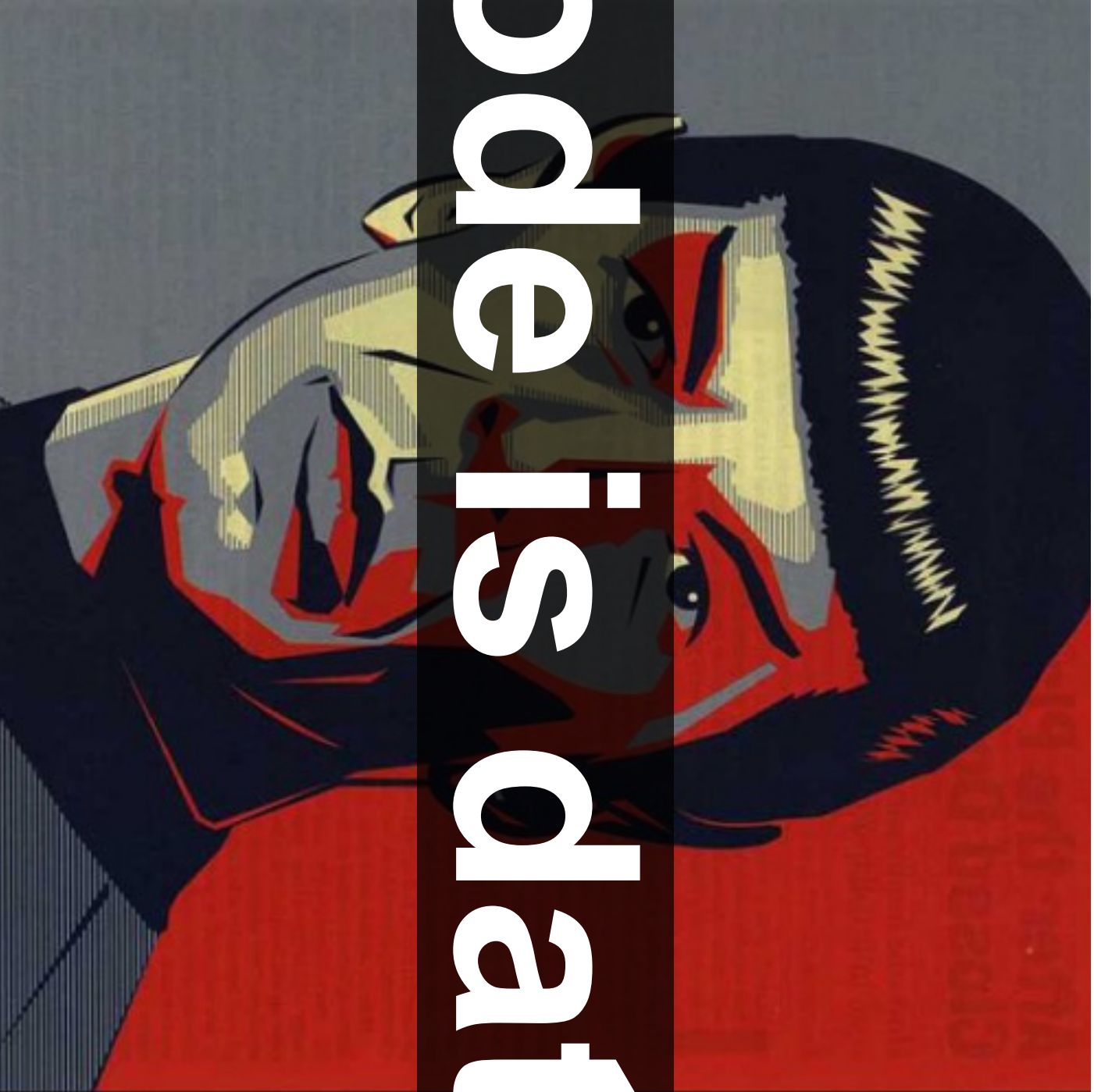


**It's life, Jim, but not as we know it!**





# Cooder is data!



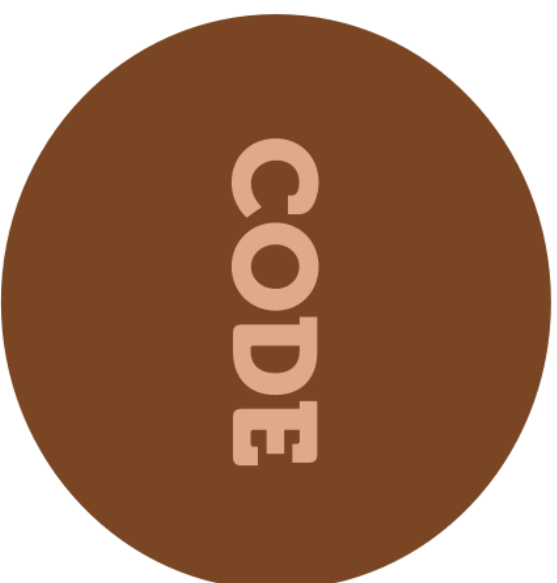


# Wvelli, big data

10

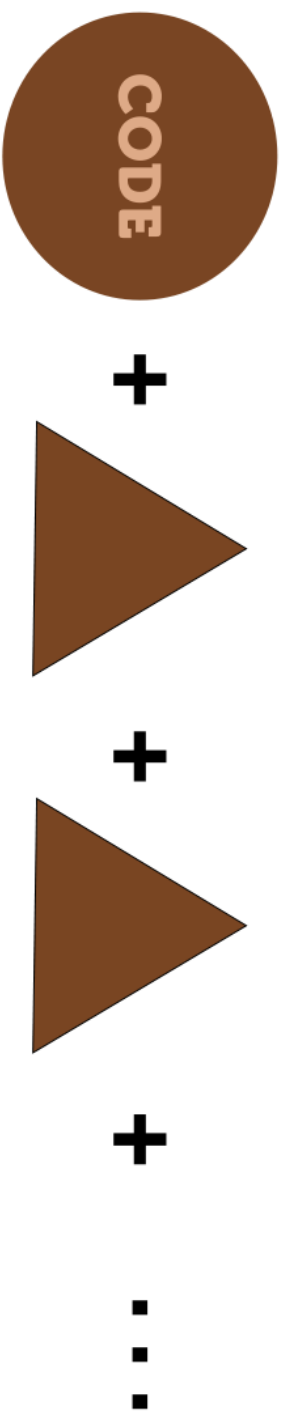


# Snapshot



statistics + metrics  
**SNAPSHOT**

# Temporal



# HISTORY

# First steps

**Count the  
lines!**



## Size does matter


- Gives you an estimate (on how much reading is needed).
- Most of the code bases are **polyglot**. Ratio between languages can tell something.
- Ratio between test code, comments, blank lines is also interesting.

# cloc



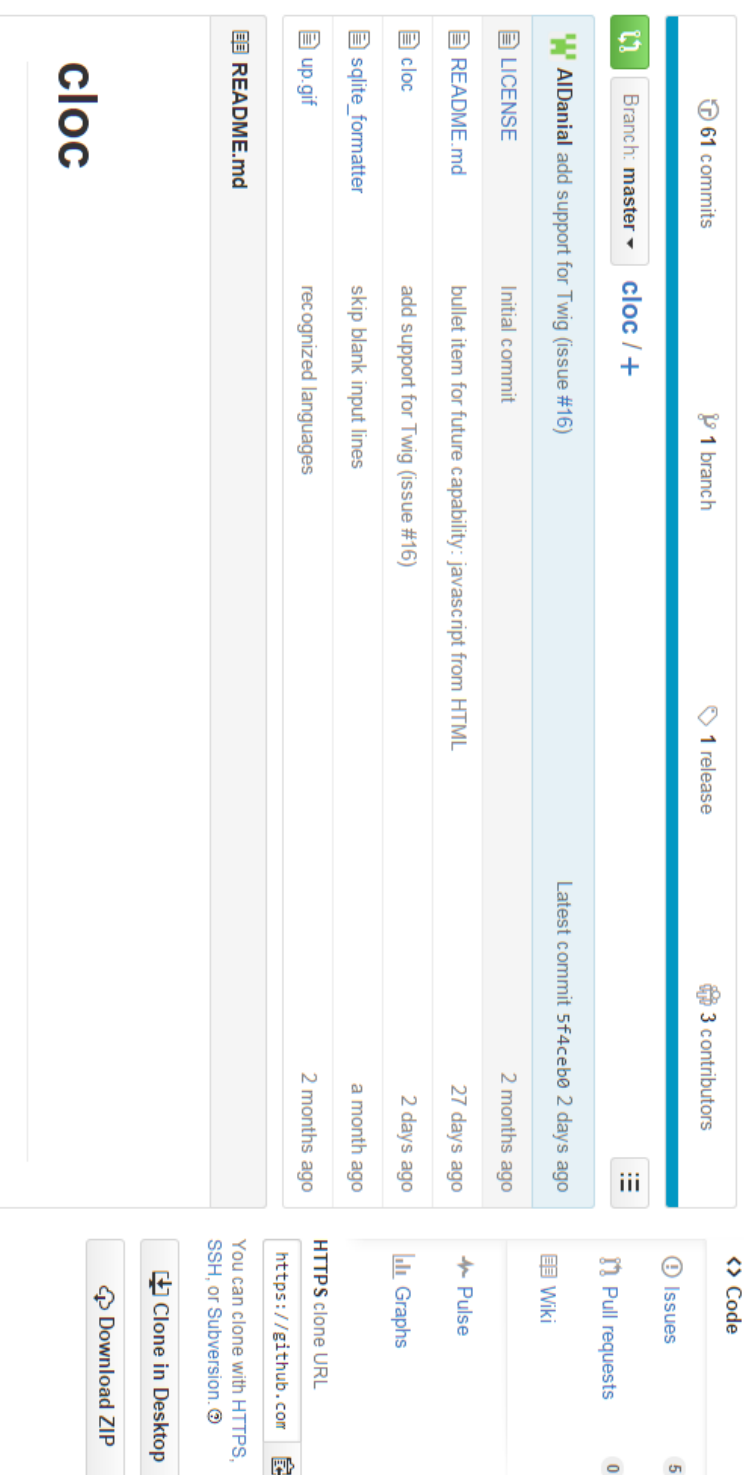
 AIDanial / cloc

 Watch **▼** 8

 Star 157

 Fork 7

cloc counts blank lines, comment lines, and physical lines of source code in many programming languages.



The screenshot shows the GitHub repository page for AIDanial/cloc. At the top, it displays 61 commits, 1 branch (master), 1 release, and 3 contributors. The latest commit is highlighted: "add support for Twig (issue #16)" by AIDanial, committed 2 days ago. Below the commit list, there are sections for LICENSE, README.md, cloc, sqlite\_formatter, up.gif, and README.md. The README.md section is expanded, showing the text "cloc". At the bottom, there are links for Code, Issues (5), Pull requests (0), Wiki, Pulse, and Graphs. The HTTPS clone URL is provided as https://github.com/AIDanial/cloc. There are also buttons for "Clone in Desktop" and "Download ZIP".





# Usage

- 01. cloc --help
- 02. cloc --write-lang-def=lang.defs



# Usage

- 01. `cloc --csv`
- 02.     `--quite`
- 03.     `--progress-rate=0`
- 04.     `--ignored=files.ignored`
- 05.     `--exclude-dir=test,build`
- 06.     `--read-lang-def=lang.defs`
- 07.     `--out=data.csv`
- 08.     `.`



# Language definitions

01. Gradle
02. `filter remove_matches ~\s*/`
03. `filter remove_inline //.*$`
04. `filter call_regex_common C`
05. `extension gradle`
06. `3rd_gen_scale 4.10`

**Where are the  
pictures?**

**We have  
stats, let's  
plot them!**

# Probably not.

## Excel?

B	C	D	E	F	G	H	I	J	K
all files	all blank	all comment	all code	files	blank	comment	code	test files	test blank
1	20	21	123	1	20	21	123	0	0
1	0	0	2	1	0	0	2	0	0
5	27	1	164	3	27	1	162	2	0
1	24	2	64	1	24	2	64	0	0
44	416	70	1022	44	416	576	192	0	0
297	5212	43	221	4	4	218	487	253	4766
8	27	2	28	2	2	0	287	0	0
7	24	0	115	5	24	0	103	2	0
6	8222	27	30413	5	7458	25586	27593	54	764
1	2	2	7	1	2	2	7	0	0
19	267	92	1562	19	267	92	1562	0	0
5	22	0	109	5	0	0	109	0	0
6	17	64	172	6	17	64	172	0	0
1	15	15	48	1	15	15	48	0	0

**Pie charts are boring!**

# Infographics!



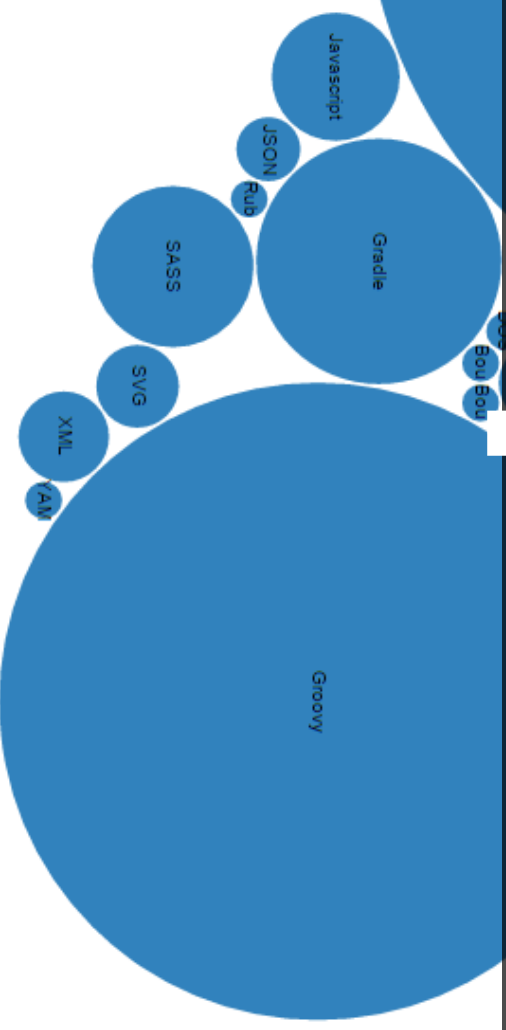




# d3.js

- Javascript library for data visualizations
- Tons of examples
- Many libraries built on top of d3.js
- Several books

# Example



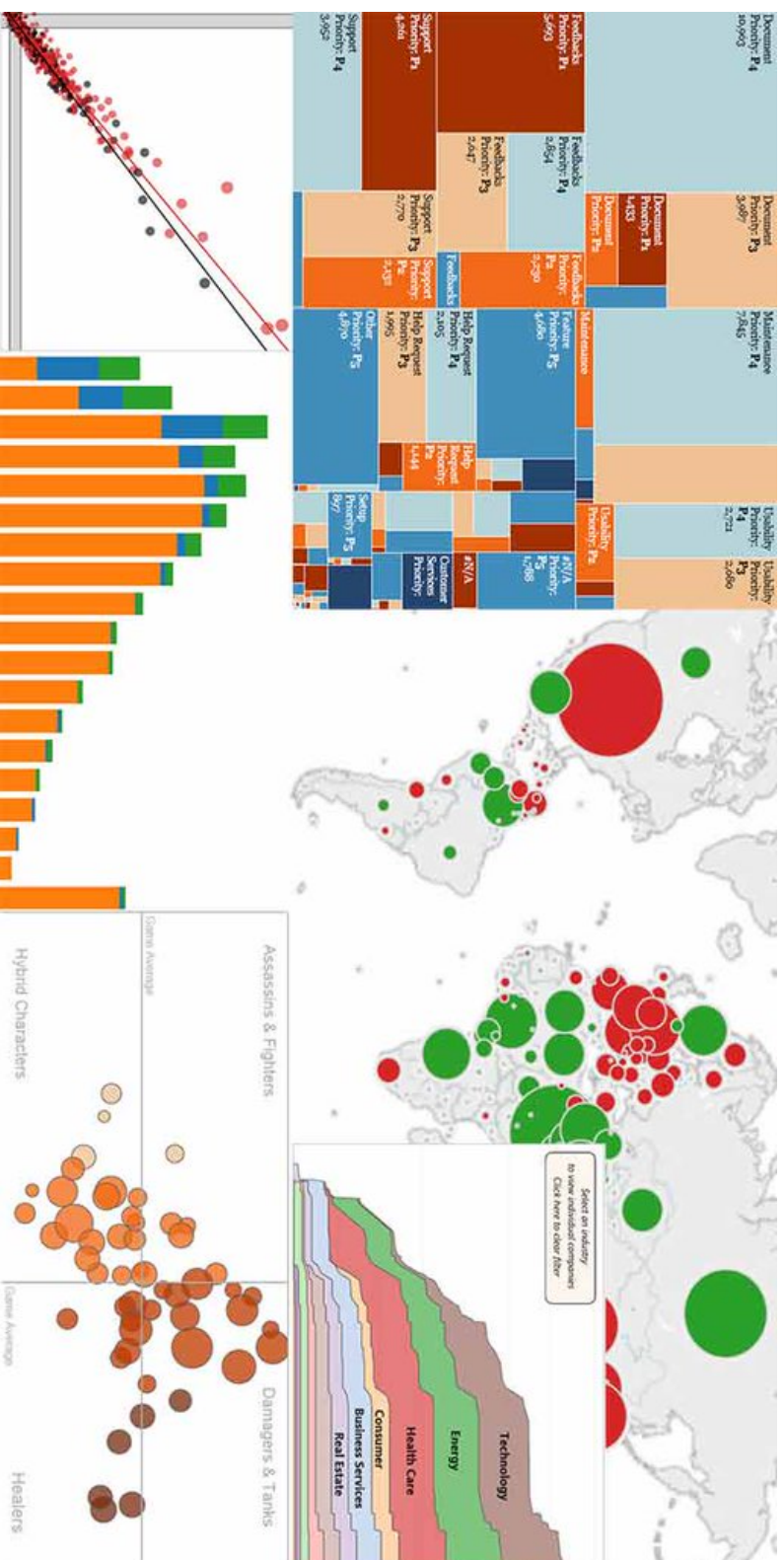
# Demo



# Many alternatives

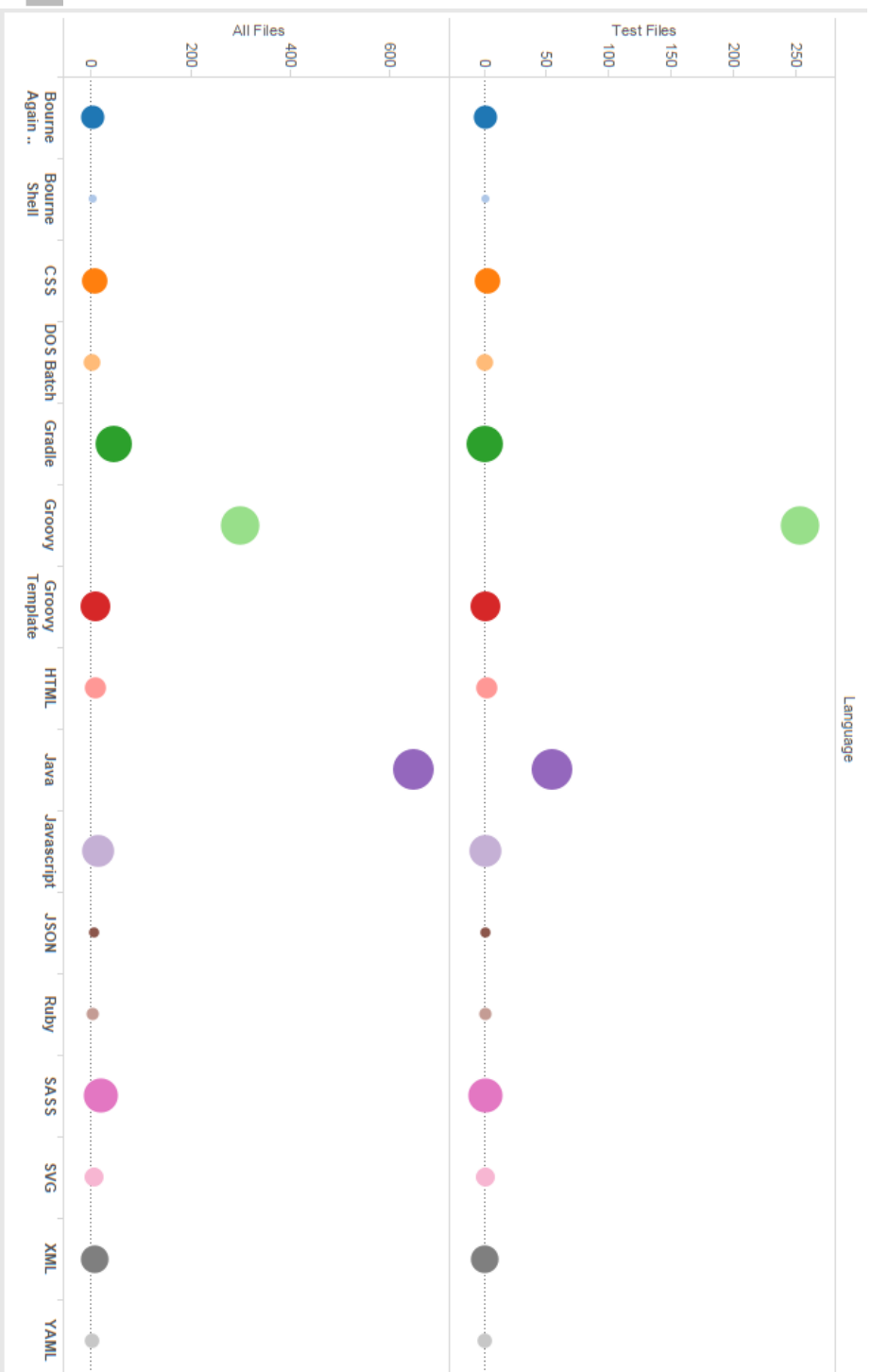
- vis.js
- raphael.js
- sigma.js
- many, many, many more

# Tableau Public



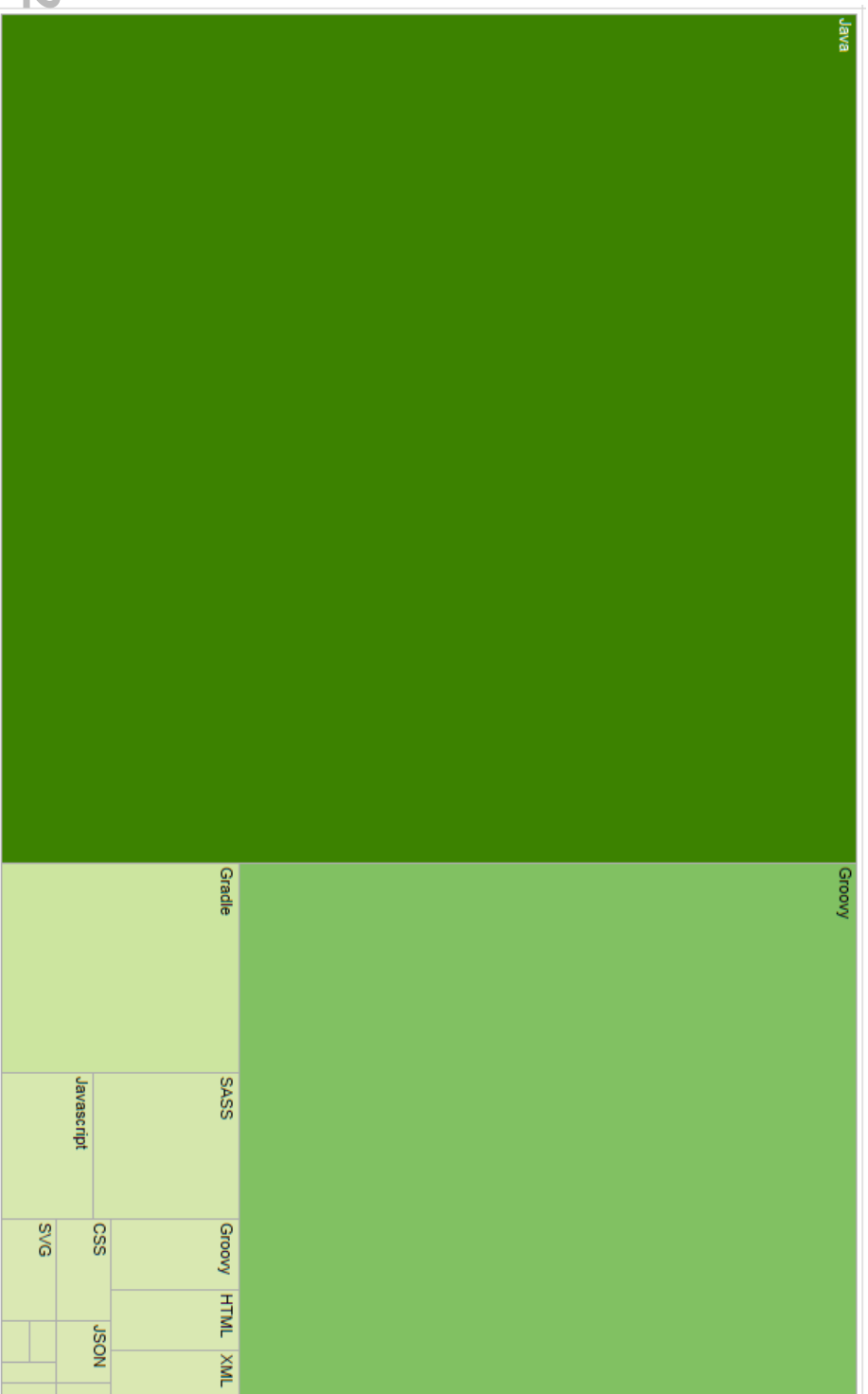


# Tableau Public





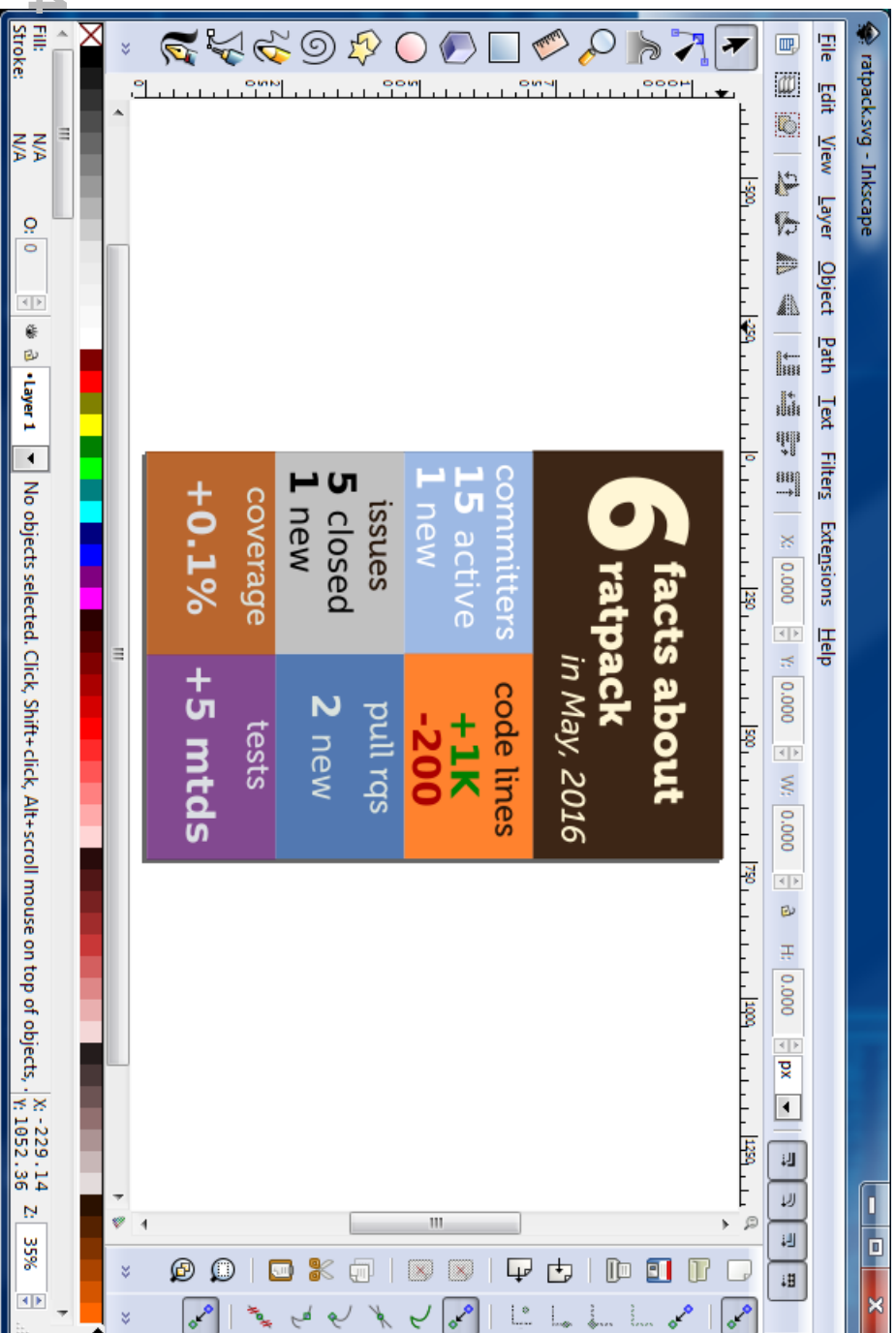
# Tableau Public





**Design your  
JUMO!**

# SVG + Inkscape



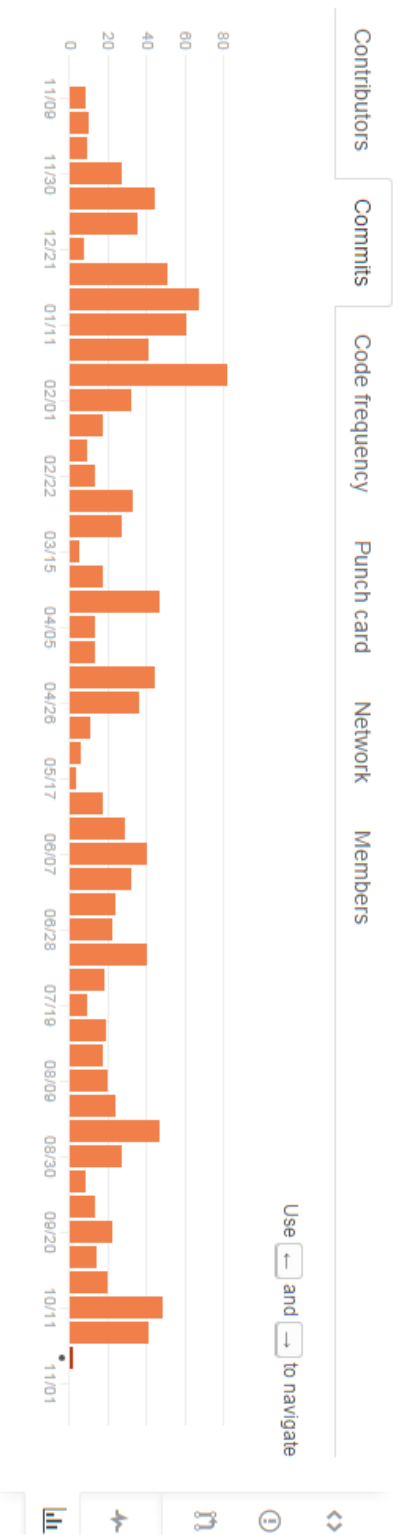


# Home-made Inforgraphics



# Temporal analysis

# GitHub



# GitHub



Contributors

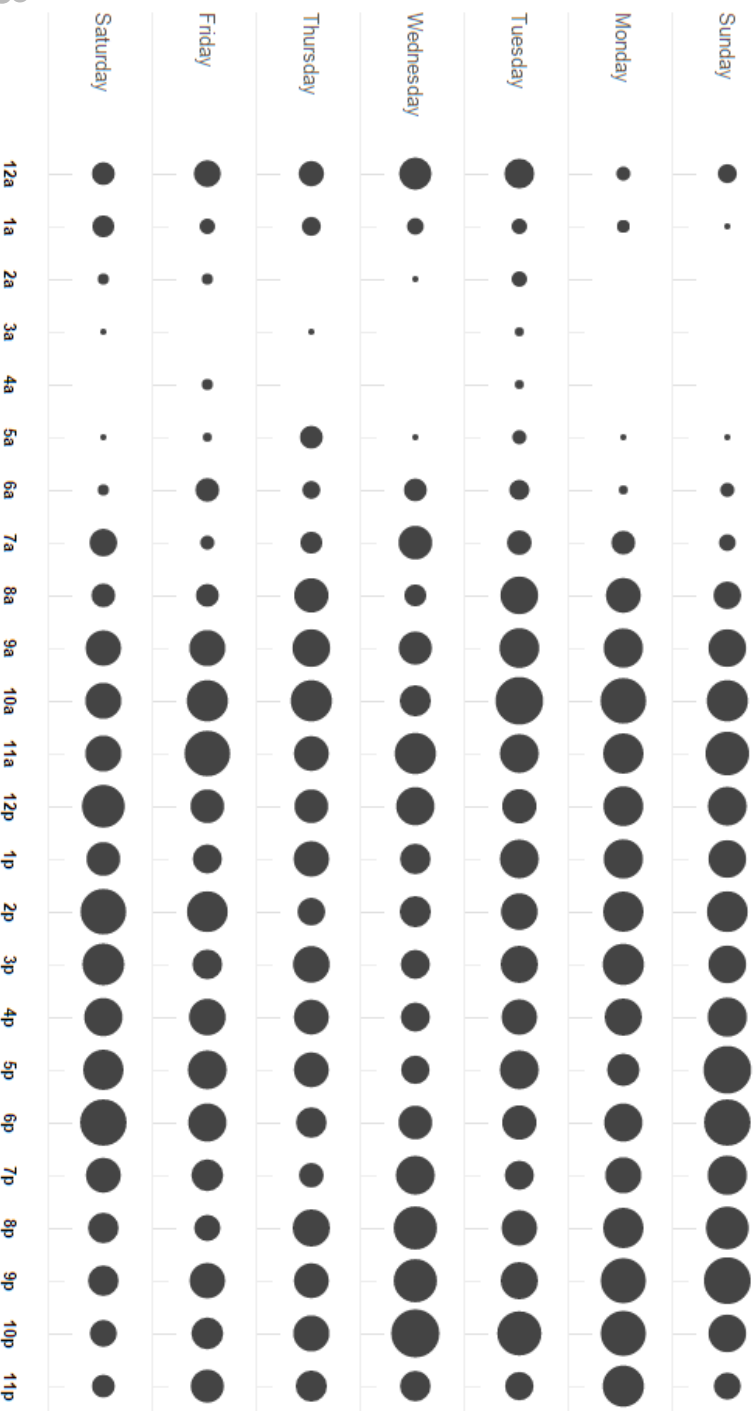
Commits

Code frequency

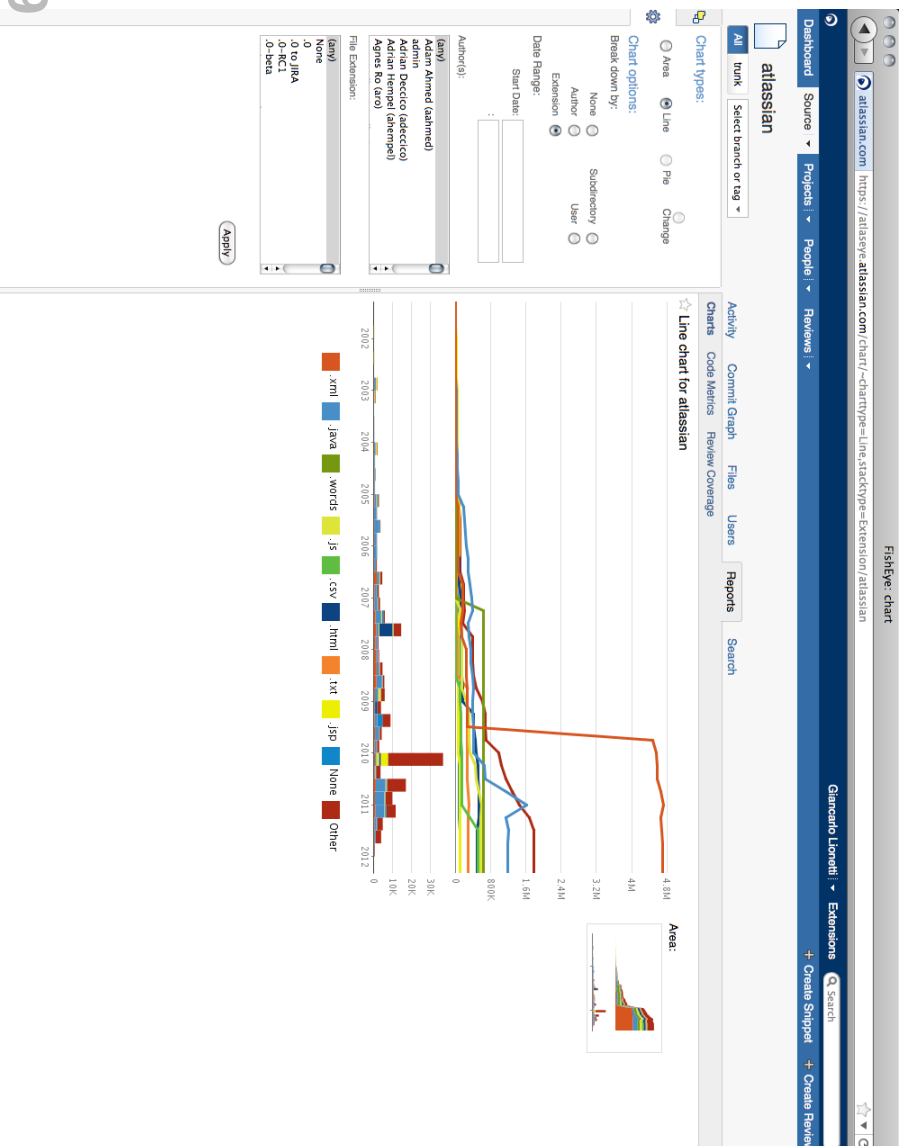
Punch card

Network

Members



# FishEye





## Gource



*Software projects are displayed by Gource as an animated tree with the root directory of the project at its centre.*

*Directories appear as branches with files as leaves.*

*Developers can be seen working on the tree at the times they contributed to the project.*





# Launch gource

- 01. gource -s 0.1 -1280x720
- 02. gource --output-custom-log log1.txt
- 03. gource -s 0.1 -1280x720 log1.txt

# Log format



- 01. 1444125624 | Luke Daley | M | /ratpack-core/.../WiretapPublisher.java
- 02. 1444125624 | Luke Daley | M | /ratpack-core/.../YieldingPublisher.java
- 03. 1444306114 | Stian Lindhom | M | /ratpack-manual/.../13-http.md
- 04. 1444312172 | Andrey Antukh | M | /ratpack-core/.../WebSocketEngine.java

# Demo



## Code Maat



*Code Maat is a command line tool used to mine and analyze data from version-control systems*



# Launch Maat

01. `git log --pretty=format: '[%h] %aN %ad %s' \`
02. `--date=short --numstat --after=YYYY-MM-DD`



# Launch Maat

01. `maat -l git.log -c git -a summary`
02. `maat -l ratpack_evo.log -c git -a revisions > ratpack_freqs.csv`
03. `python scripts/merge_comp_freqs.py ratpack_freqs.csv ratpack_lines`



# Analysis types

abs-churn, age, author-churn, authors, communication, coupling, entity-churn, entity-effort, entity-ownership, fragmentation, identity, main-dev, main-dev-by-revs, messages, refactoring-main-dev, revisions, soc, summary

# Demo





# Let's talk big data





# ElasticSearch

- Distributed, scalable, and highly available
- Real-time search and analytics capabilities
- Sophisticated RESTful API



## Index data

```
01. $ curl -XPUT 'http://localhost:9200/gitlog/commit/123345' -d '{
02.   "commitId" : "123345",
03.   "timestamp" : "2009-11-15T14:12:12",
04.   "message" : "git into es"
05. }'
```



# Kibana

- Flexible analytics and visualization platform
- Real-time summary and charting of streaming data
- Intuitive interface for a variety of users
- Instant sharing and embedding of dashboards



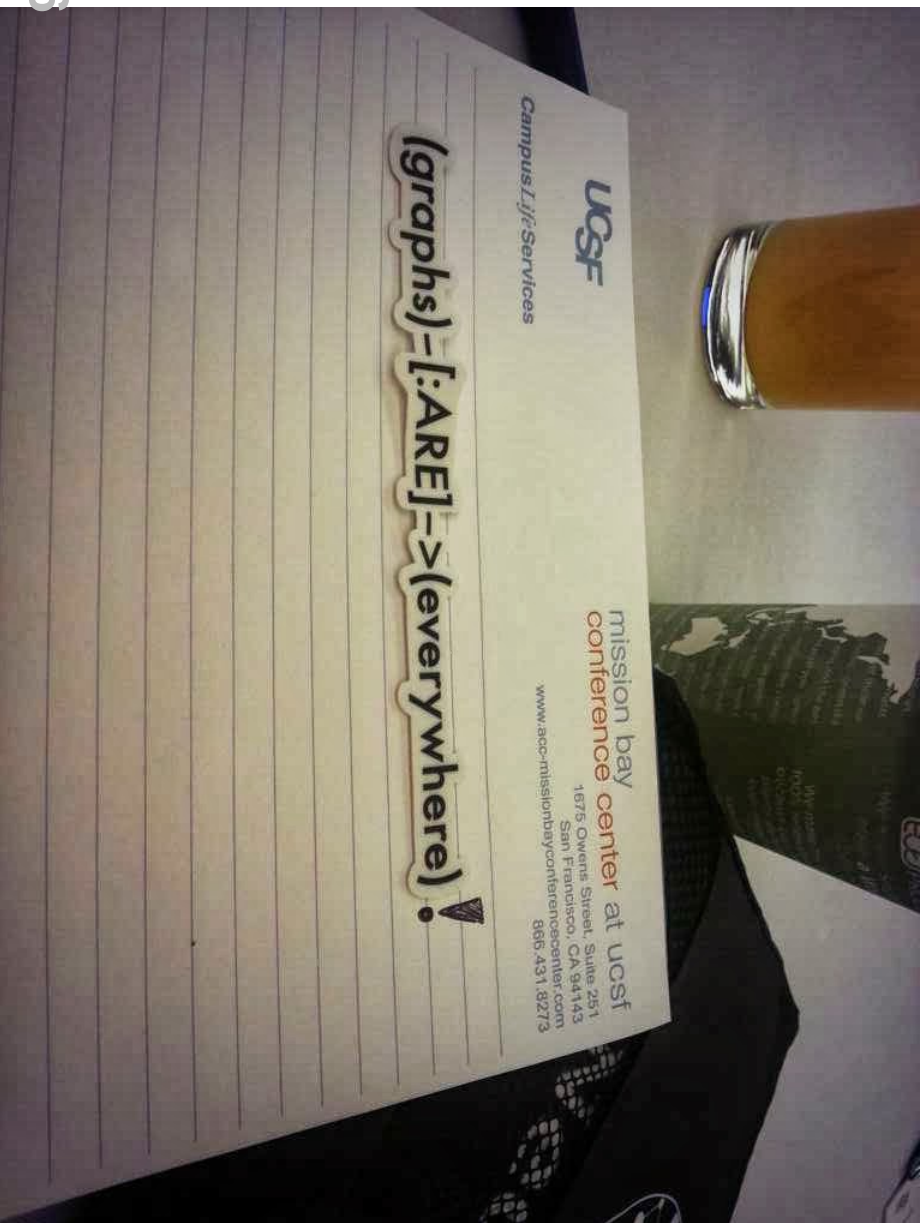


# Tag Cloud

01. kibana plugin -i tagcloud -u <https://github.com/stormpython/tagcloud>

# Demo

# Graphs are everywhere!







# jQAssistant



*jQAssistant is a QA tool which allows the definition and validation of project specific rules on a structural level.*

*It is built upon the graph database Neo4j and can easily be plugged into the build process to automate detection of constraint violations and generate reports about user defined concepts and metrics.*



# jQAssistant

01. jqassistant scan -f binaries
02. jqassistant server



# Query your code

01. MATCH
02. (class:Class) -[:DECLARES] ->(method:method)
03. RETURN
04. class.fqn, count(method) as Methods
05. ORDER BY
06. Methods DESC
07. LIMIT 20

# neo4j



5

☆ + | ▾

```
CYPHER MATCH (e:Type)-[:DECLARES]->(init:Constructor) WHERE e.fqn='java.io.IOException' WITH
```

Method  
File  
Type

Displaying 17 nodes, 33 relationships

⌵

# Demo

**That's it!**



# Conclusion

- Extract data from your code!
- Visualize it!
- Search for new facts and knowledge!
- Become data scientist or data journalist!

# Reading material





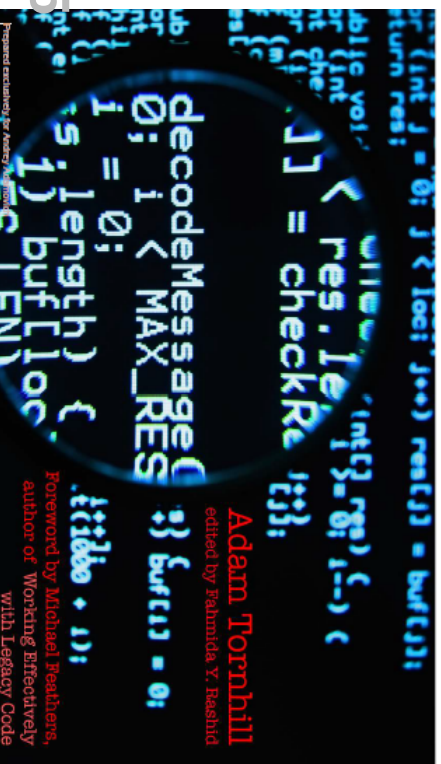
# Your Code as a Crime Scene

The  
Pragmatic  
Programmers

Investigator: \_\_\_\_\_  
Date: \_\_\_\_\_  
Case #: \_\_\_\_\_  
Location: \_\_\_\_\_

## Your Code As a Crime Scene

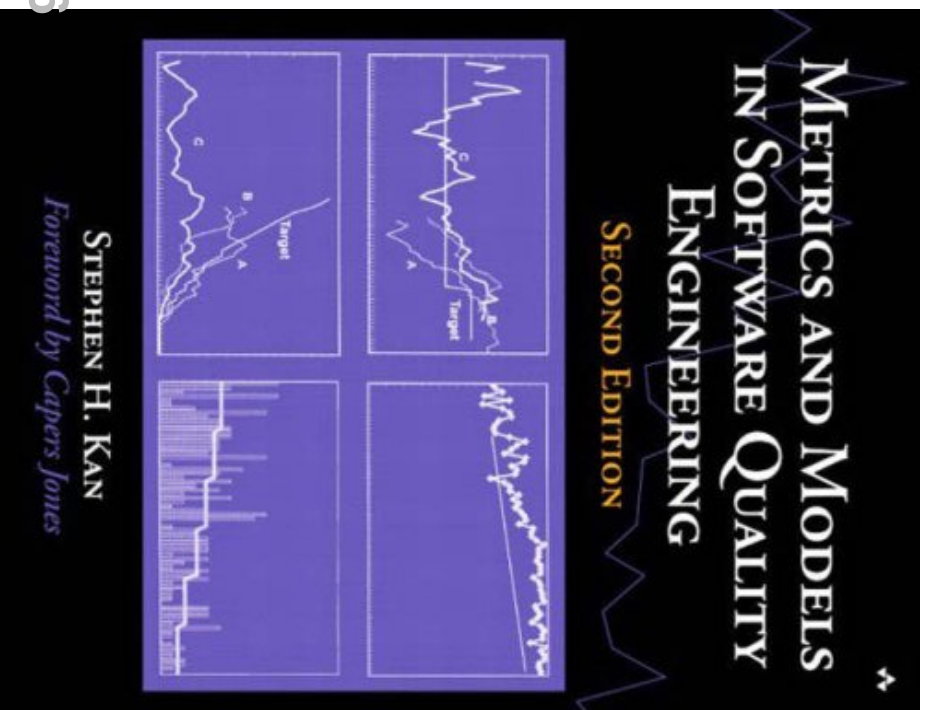
Use Forensic Techniques  
to Arrest Defects, Bottlenecks, and  
Bad Design in Your Programs



65



# Metrics and Models in SQE

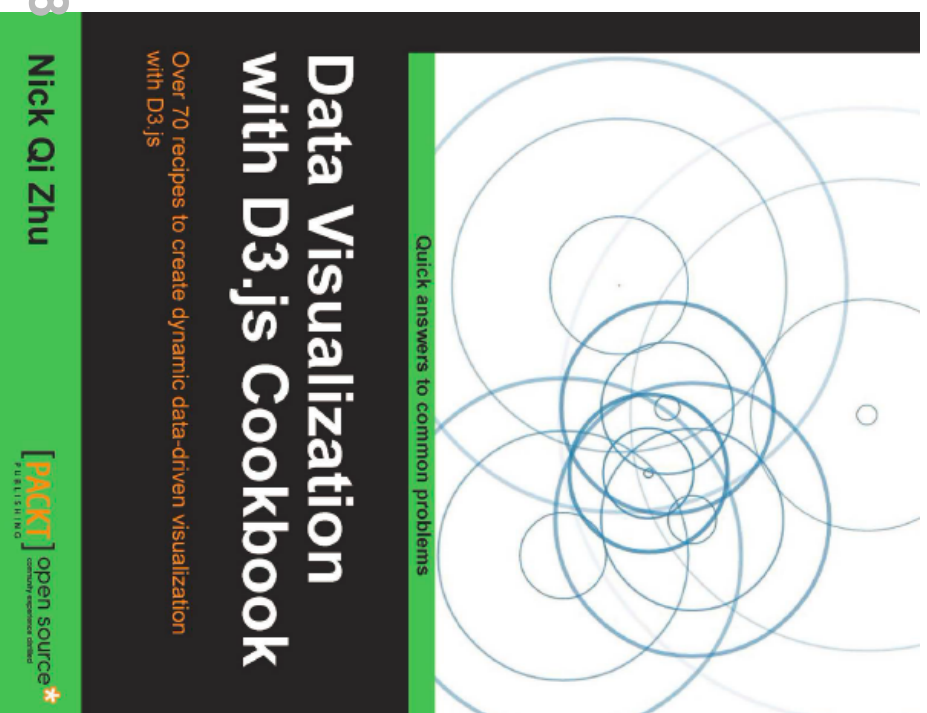


# Software Metrics and Metrology





# D3 Cookbook



# D3 Tips & Tricks





# Links: images

- <http://abstrusegoose.com/432>
- <http://camarenaphoto.tumblr.com/post/112238079516/its-life-jim-but-not-as-we-know-it-spock>
- <http://technology.ie/big-data-looks-like/>
- <http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>
- <http://github.info/>
- <http://emmanueloga.com/2013/10/07/Graphs-are-Everywhere-An-overview-of-GraphConnect-San-Francisco-2013.html>

70



# Links : tools

- <https://github.com/AIDanial/cloc>
- <http://gource.io/>
- <http://wettel.github.io/codecity-wof.html>
- <https://github.com/adamtorhill/code-maat>
- <http://d3js.org/>
- <http://visjs.org/>



## Links : tools

- <http://www.sonarqube.org/>
- <https://www.atlassian.com/pt/software/fisheye/overview>
- <https://www.elastic.co/products/elasticsearch>
- <https://www.elastic.co/products/kibana>
- <http://jqassistant.org/>
- <http://neo4j.com/>





## Links : tools

- [https://github.com/ThoughtWorksStudios/saikuro\\_treemap](https://github.com/ThoughtWorksStudios/saikuro_treemap)
- <https://www.youtube.com/watch?v=iilytERhV9o>

**Thank you!**

# Questions?