

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks

Software Development Analytics with GrimoireLab

Jesus M. Gonzalez-Barahona

Universidad Rey Juan Carlos

@jgbarah <http://github.com/jgbarah/presentations>

Intl. Summer School on Visual Soft. Analytics
Leipzig (Germany), September 23rd 2019

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

*It is difficult to improve
if you cannot measure
and track your improvement*

Our plan today

- ① A bit of context
- ② Dealing with dynamic complexity
- ③ Data sources
- ④ GrimoireLab
- ⑤ Case studies

Activity

Remaining code

Performance

Demographics

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

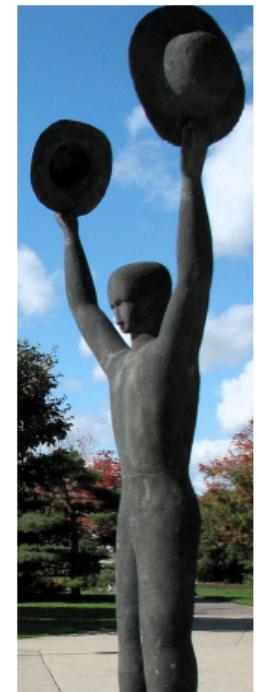
Final remarks

Me and my two hats

Uni Rey Juan Carlos:

- Understanding free, open source software
- Data analytics approach
- Data visualization in XR

<http://gsyc.es/jgb>



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

Me and my two hats

Bitergia:

- From research to the real world
- Understanding software development
- Data analytics approach

<http://bitergia.com>



Recommendations

- Open your laptop
- Download the slides (they have links)
- Visit Alpha.Cauldron.io and produce your own dashboard
- Play with the dashboards
- Understand the interpretations behind the numbers

<https://alpha.cauldron.io>

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks

Cauldron Alpha

The screenshot shows the Cauldron Alpha interface for the Django project. At the top, there's a header with the project name "Django" and a "View project data" button. Below the header, a section titled "Add data sources" lists several options: GitHub, Git, GitLab, Meetup, and a note about future data source additions. A table then displays 36 data sources found, with columns for Status, Data source, Last refresh, and Duration. Each row includes links for Logs, Delete, and Refresh.

Status	Data source	Last refresh	Duration	Logs	Delete	Refresh
✓	https://github.com/django/ticketbot.git	a month ago	00:00:17	Logs	Delete	Refresh
✓	https://github.com/django/djangonippets.org.git	a month ago	00:00:25	Logs	Delete	Refresh
✓	https://github.com/django/djangoproject.com.git	a month ago	00:00:47	Logs	Delete	Refresh
✓	https://github.com/django/djangobench.git	a month ago	00:00:21	Logs	Delete	Refresh
✓	https://github.com/django/django-localflavor.git	a month ago	00:00:36	Logs	Delete	Refresh
✓	https://github.com/django/django-formtools.git	a month ago	00:00:24	Logs	Delete	Refresh
✓	https://github.com/django/django-docs-translations.git	a month ago	00:00:33	Logs	Delete	Refresh
✓	https://github.com/django/django-contrib-comments.git	a month ago	00:00:21	Logs	Delete	Refresh
✓	https://github.com/django/django-box.git	a month ago	00:00:17	Logs	Delete	Refresh
✓	https://github.com/django/django.git	a month ago	00:11:09	Logs	Delete	Refresh
✓	https://github.com/django/deps.git	a month ago	00:00:20	Logs	Delete	Refresh

It is planned for the future to add more data sources (like Discourse or Slack), but feel free to suggest any other data source option via the feedback button!

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

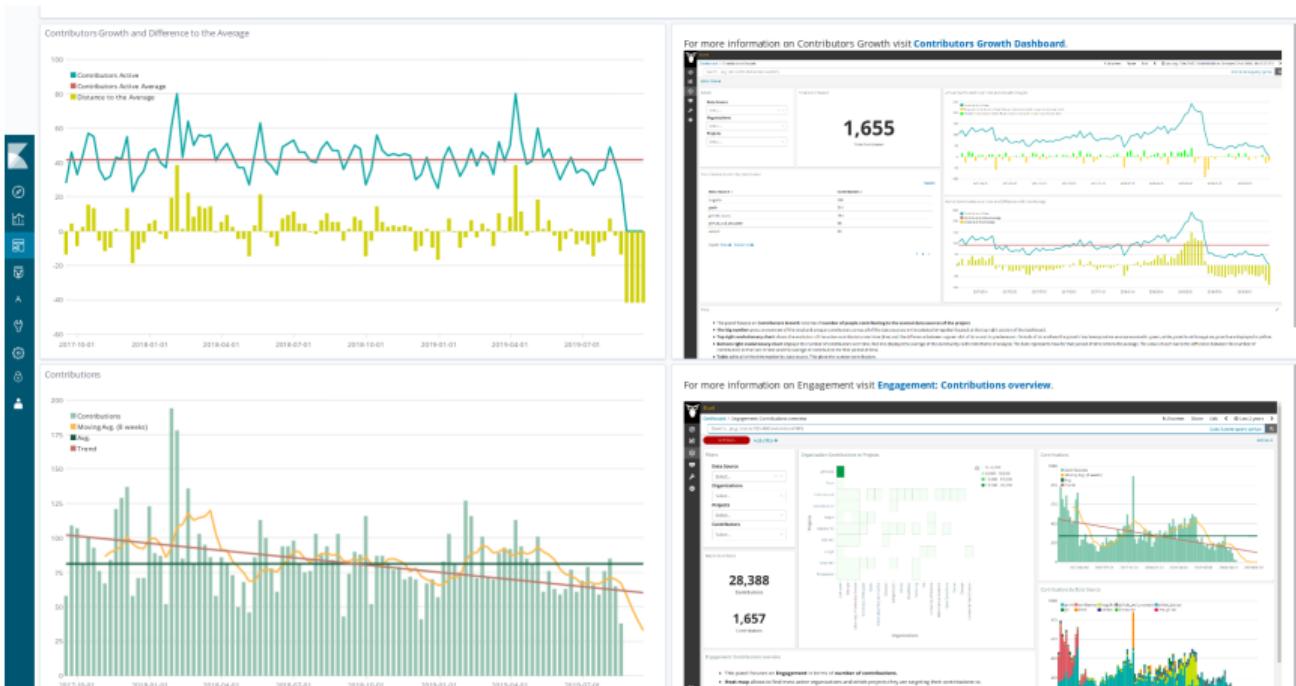
Performance

Demographics

Diversity

Final remarks

Cauldron Alpha



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

Development projects may be large and complex



Projects may be large and complex...
and dynamic

It's difficult to...

- ...track what's happening
- ...understand why it's happening
- ...react quickly
- ...evaluate results of reaction

If data is available
analytics may come to the rescue

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

A continuous process

Figure out your interest

Find out available data

Define key parameters

Monitor, understand, detect deviations

Act to correct, improve

Track results

Measure → Monitor → Act

A continuous process

Case example: Overall development activity

Interest: activity

Data: changes to code, tickets

Parameters: commits, tickets closed

Monitoring: charts, numbers

Observation: numbers declining

Action: allocate more developer effort

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

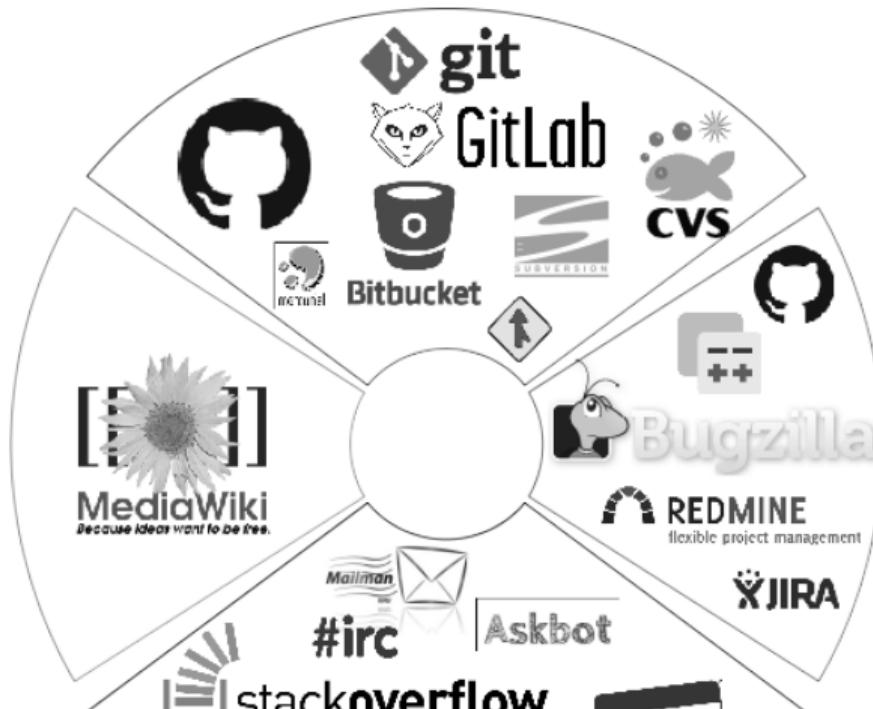
GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

Repositories, repositories...



Source code management

- Client/server: CVS, Subversion
- Decentralized: git, Mercurial, Bazaar, etc.
- Most of them accessible through git...
(with some problems)
- Can be integrated with other tools:
Gerrit, GitHub, GitLab, etc.

Issue tracking

Many different systems:

- Bugzilla
- Jira
- GitHub issues
- GitLab Issues
- Phabricator
- RedMine...

Each with a different model, data, operations...

Code review

Usually: peer review pre-merge review

Different methods:

- Mailing lists (eg: Linux)
- Gerrit (eg: OpenStack)
- GitHub pull requests (eg: ElasticSearch)
- GitLab merge requests (eg: GNOME)
- or even Jira, Bugzilla...

Much of the control on the software lies here

Async communication

Mailing lists:

- Mailing lists systems (Mailman)
- Google Groups
- Mailing list archivers

Forums: too many to mention

Question/Answer sites: StackOverflow, Askbot

Information is always archived

Sync communication

Systems:

- Traditionally: IRC
- Nowadays: Slack & many others
- Not always text/based (eg: videoconferences)

Notes:

- In many cases, lack of archives
- Privacy concerns: considered informal

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

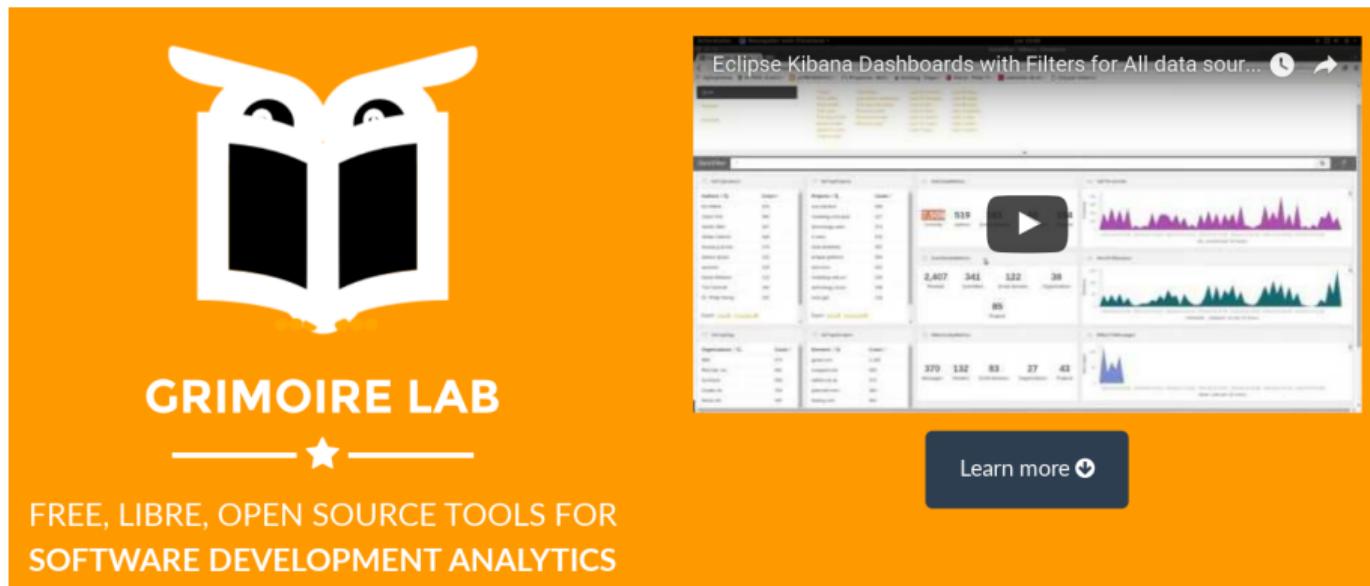
Remaining code

Performance

Demographics

Diversity

Final remarks



The image shows the GrimoireLab website landing page on the left and a screenshot of the Eclipse Kibana dashboard on the right. The landing page features a large orange background with a white owl logo and the text 'GRIMOIRE LAB'. Below it, a star icon is flanked by two horizontal lines. The text 'FREE, LIBRE, OPEN SOURCE TOOLS FOR SOFTWARE DEVELOPMENT ANALYTICS' is displayed. A 'Learn more' button is visible. The dashboard screenshot shows various data visualizations and filters.

<https://chaoss.github.io/grimoirelab>

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

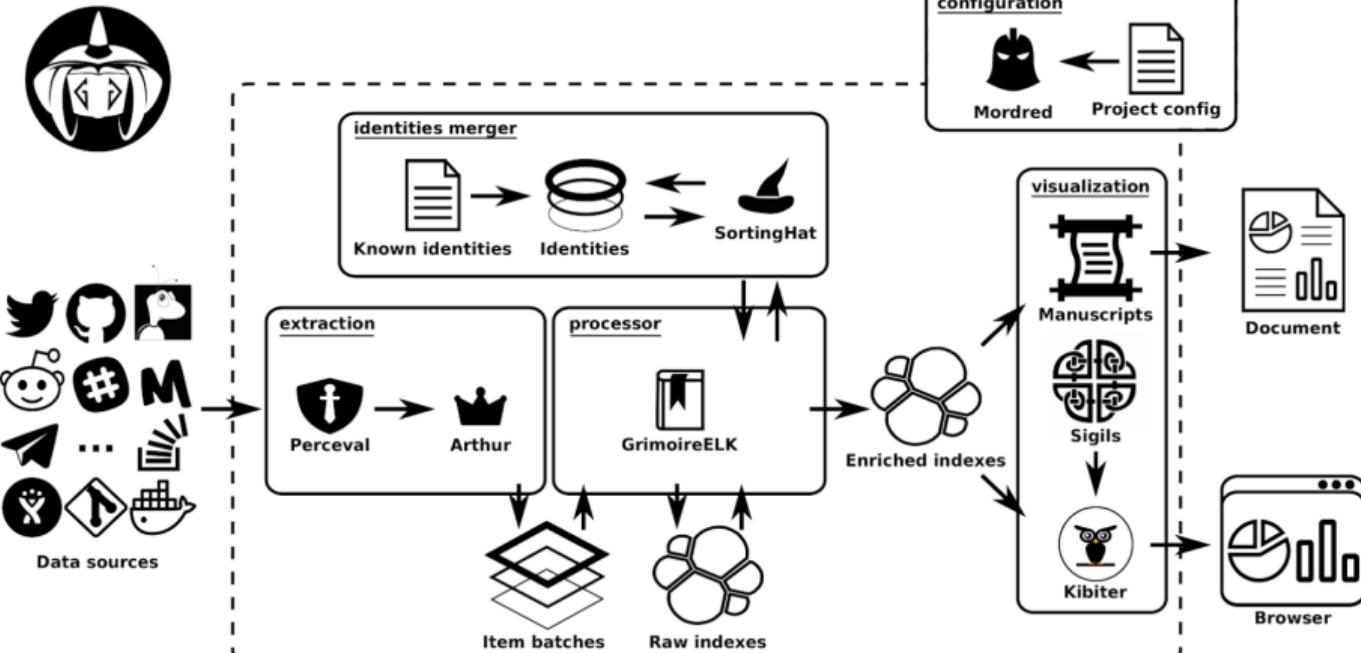
Data sources

GrimoireLab

Case studies

- Activity
- Remaining code
- Performance
- Demographics
- Diversity

Final remarks



<https://chaoss.github.io/grimoirelab>

Main components

- Perceval: data retrieval
- Arthur: retrieval orchestration
- GelK: enrichment
- SortingHat: identity management
- ElasticSearch (*): database
- Kibiter: dashboard (light fork of Kibana)
- Sigils: visualizations for Kibana/Kibiter

(*) Not a part of GrimoireLab

Perceval

```
$ python3 -m venv gl
$ source gl/bin/activate
(gl) $ pip install grimoirelab
(gl) $ perceval git \
      https://github.com/chaoss/grimoirelab-perceval
(gl) $ perceval github \
      chaoss grimoirelab-perceval
```

<https://chaoss.github.io/grimoirelab-tutorial/perceval>

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks

```
{"backend_name": "Git",
"backend_version": "0.11.1",
"category": "commit",
"classified_fields_filtered": null,
"data": {
    "Author": "Santiago Due\u00f1as <sduenas@bitergia.com>",
    "AuthorDate": "Tue Aug 18 18:08:27 2015 +0200",
    "Commit": "Santiago Due\u00f1as <sduenas@bitergia.com>",
    "CommitDate": "Tue Aug 18 18:08:27 2015 +0200",
    "commit": "dc78c254e464ff334892e0448a23e4cfbf637a3",
    "files": [
        "action": "A",
        "added": "10",
        "file": ".gitignore",
```

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks

```
{"backend_name": "GitHub",
"backend_version": "0.22.1",
"category": "issue",
"classified_fields_filtered": null,
"assignee_data": {},
"assignees": [],
"assignees_data": [],
"author_association": "CONTRIBUTOR",
"body": "Based on Sphynx, prepared...",
"closed_at": "2016-01-04T13:51:56Z",
"comments": 0,
"comments_data": [],
"comments_url": "https://api.github.com/...",
"created_at": "2016-01-03T23:46:04Z",
```

Perceval as a module

```
#! /usr/bin/env python3
from perceval.backends.core.git import Git

repo_url = 'http://github.com/chaos/grimoirelab-perceval'
repo_dir = '/tmp/perceval.git'

repo = Git(uri=repo_url, gitpath=repo_dir)
for commit in repo.fetch():
    print(commit['data']['commit'])
```

```
import argparse
from perceval.backends.core.git import Git

parser = argparse.ArgumentParser(description = "Count commits")
parser.add_argument("repo", help = "Repository url")
parser.add_argument("--print", action='store_true', help = "Print commits")
args = parser.parse_args()

repo = Git(uri=args.repo, gitpath='/tmp/perceval.git')
count = 0
for commit in repo.fetch():
    if args.print:
        print(commit['data']['commit'])
    count += 1
print("Number of commits: %d." % count)
```

SirMordred

Producing a dashboard:

- Elasticsearch installed
- Kibana / Kibiter installed
- MariaDB installed
- Config: mordred.cfg, projects.json, identities.yaml, menu.yaml

```
(gl) $ mordred -c mordred.cfg
```

<https://chaoss.github.io/grimoirelab-tutorial/sirmordred>

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

GrimoireLab



Software development analytics with
free, open source software



(a CHAOSS project)

chaoss.github.io/grimoirelab
chaoss.github.io/grimoirelab-tutorial

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks



Tracking involved parties

Development is much more than developers
(this is explicit in FOSS & inner sourcing)

- Developers: all repositories
- Contributors: issue tracking, async communication
- Users: async communication, ...
- Ecosystem: difficult to track

Activity / size

- committing patches:
source code management system
- reporting, commenting or fixing bugs:
issue tracking system
- submitting patches or reviewing them:
code review system
- sending messages:
async or sync communication systems

Most common cases

- Parameters reflecting activity for a period.
- People active for a certain period.
- Evolution of any of them.
- Trends for any of them.

Difficult to compare between projects
Interesting to compare in-project

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks

Many facets



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

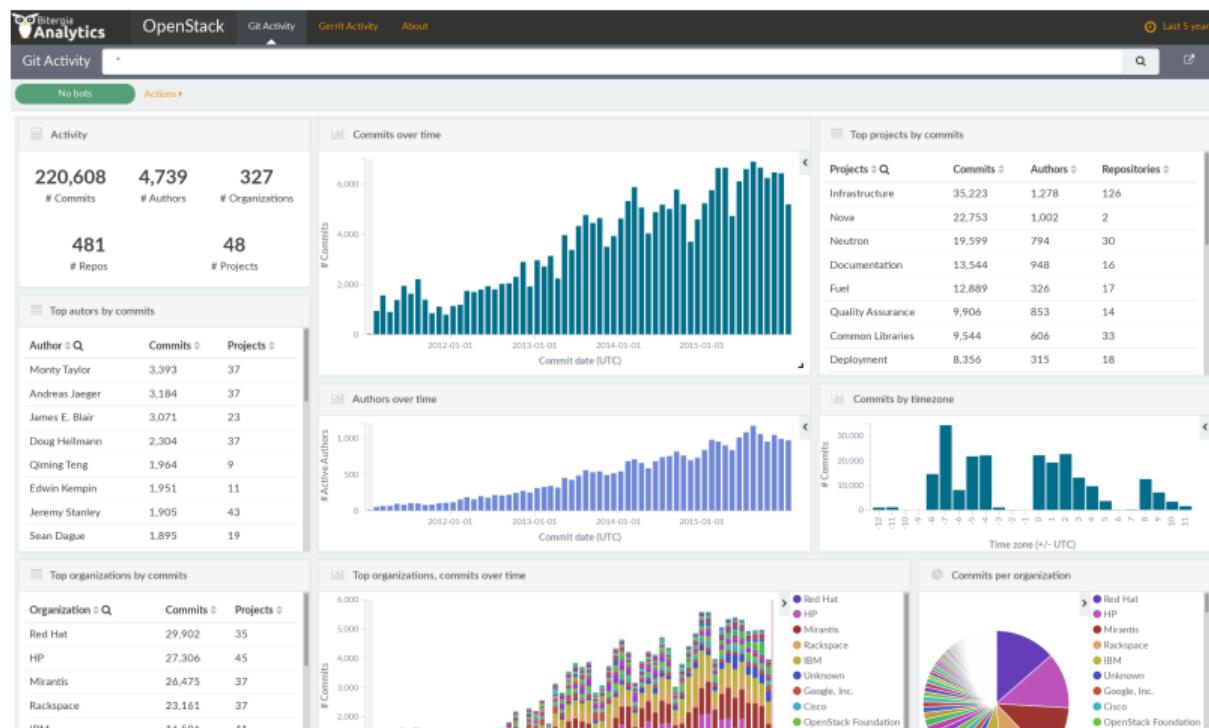
Remaining code

Performance

Demographics

Diversity

Final remarks



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

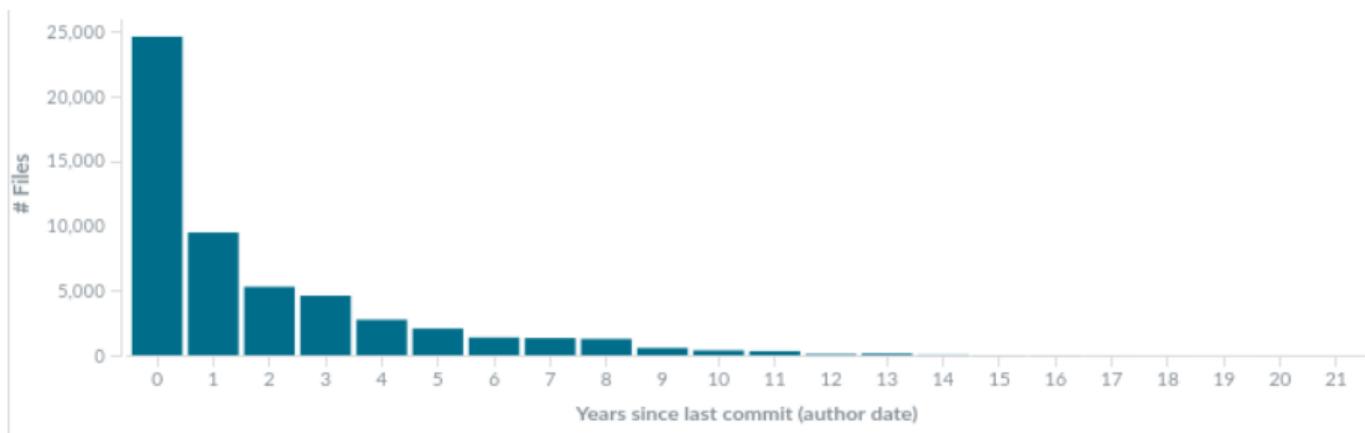
Performance

Demographics

Diversity

Final remarks

How old is code?



[Linux kernel, July 2016, C files by last commit]

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

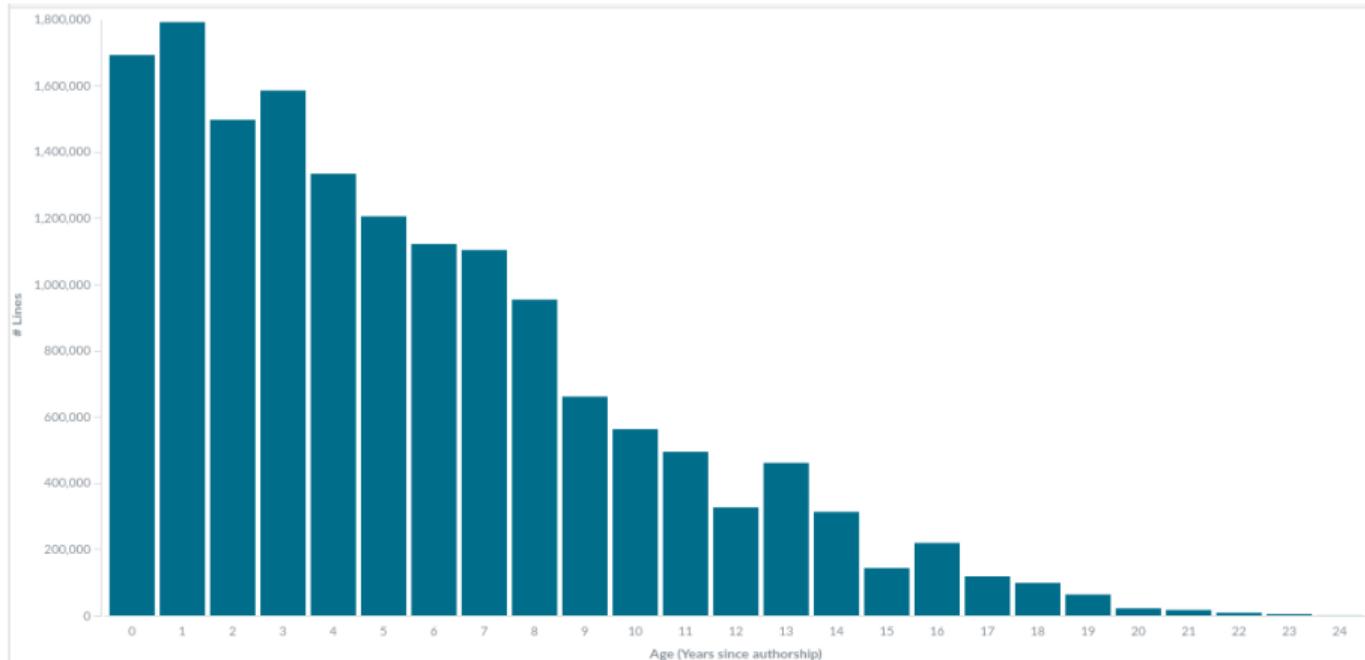
Remaining code

Performance

Demographics

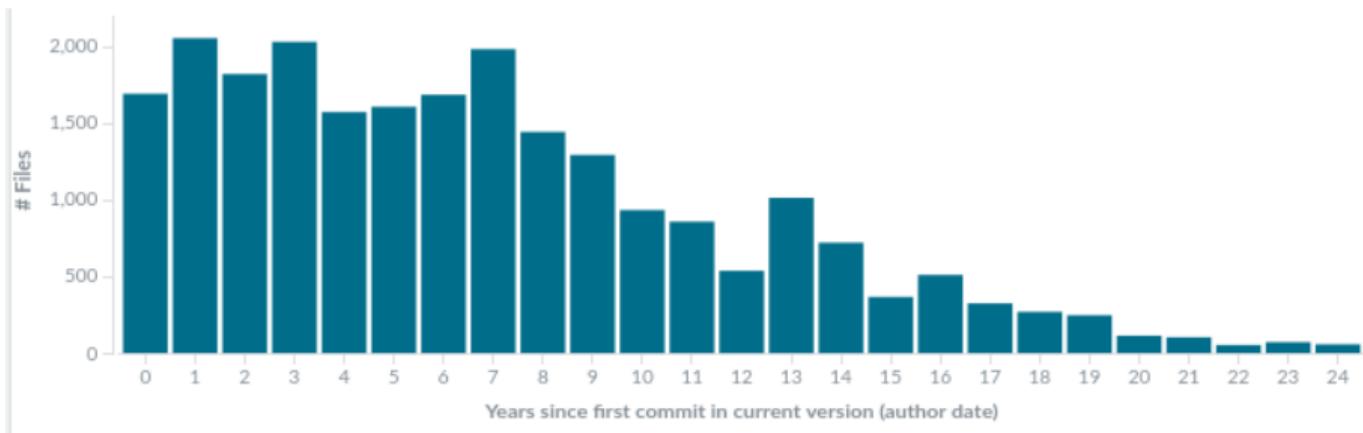
Diversity

Final remarks



[Linux kernel, July 2016, lines in C files by age]

How old is code (3)



[Linux kernel, July 2016, C files by first
remaining commit]

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

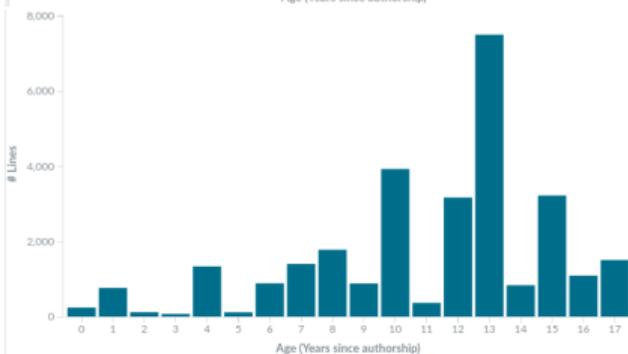
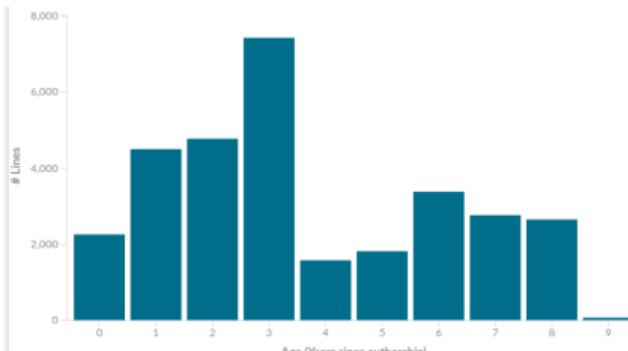
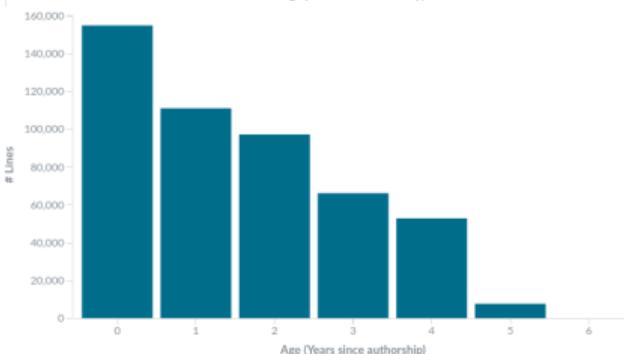
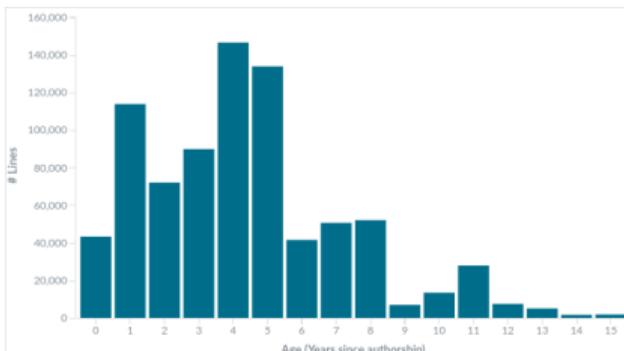
Remaining code

Performance

Demographics

Diversity

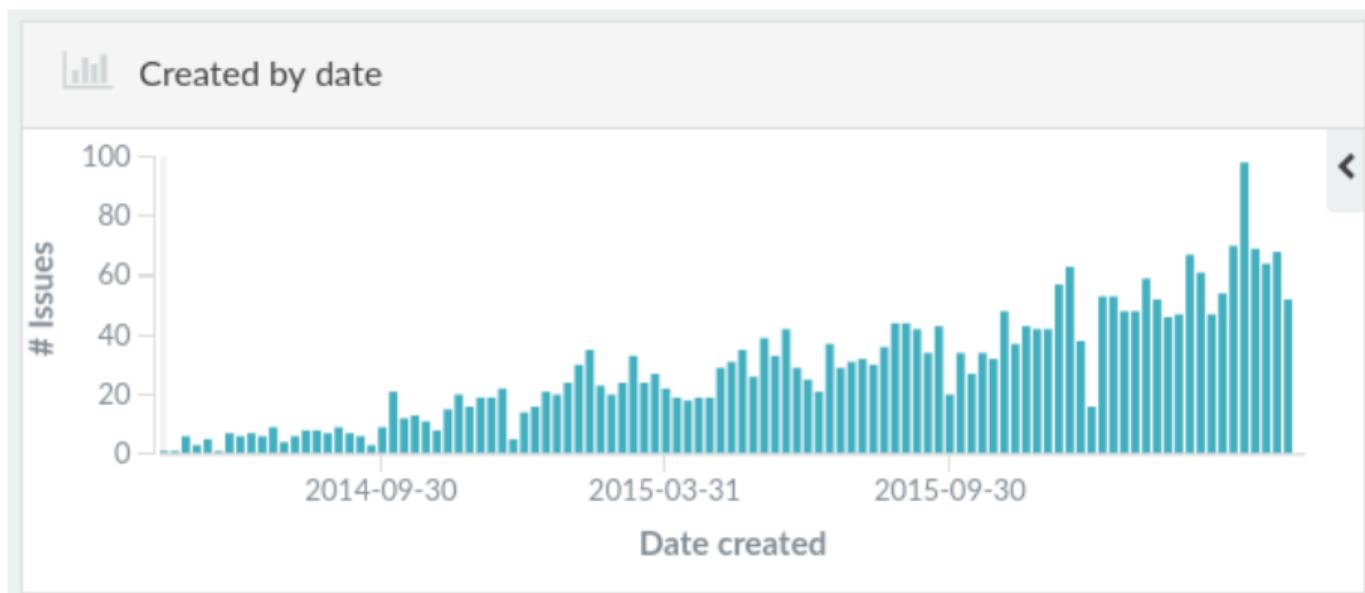
Final remarks



Age of lines (data of authorship, “.c” files in Linux)

From top left, clockwise: Wireless, USB, IRDA Ethernet

Performance (backlog)



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

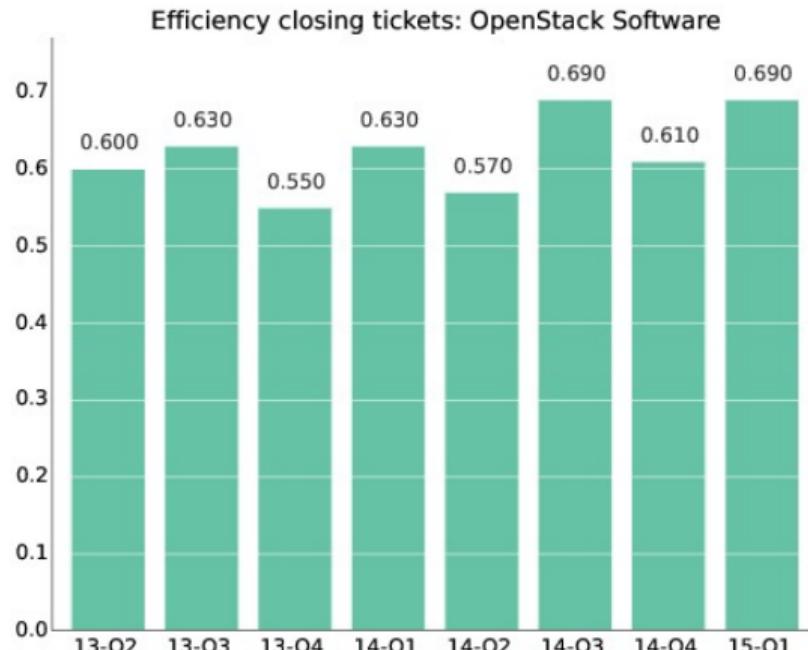
Remaining code

Performance

Demographics

Diversity

Final remarks



Efficiency. Example: closed/opened tickets per quarter

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

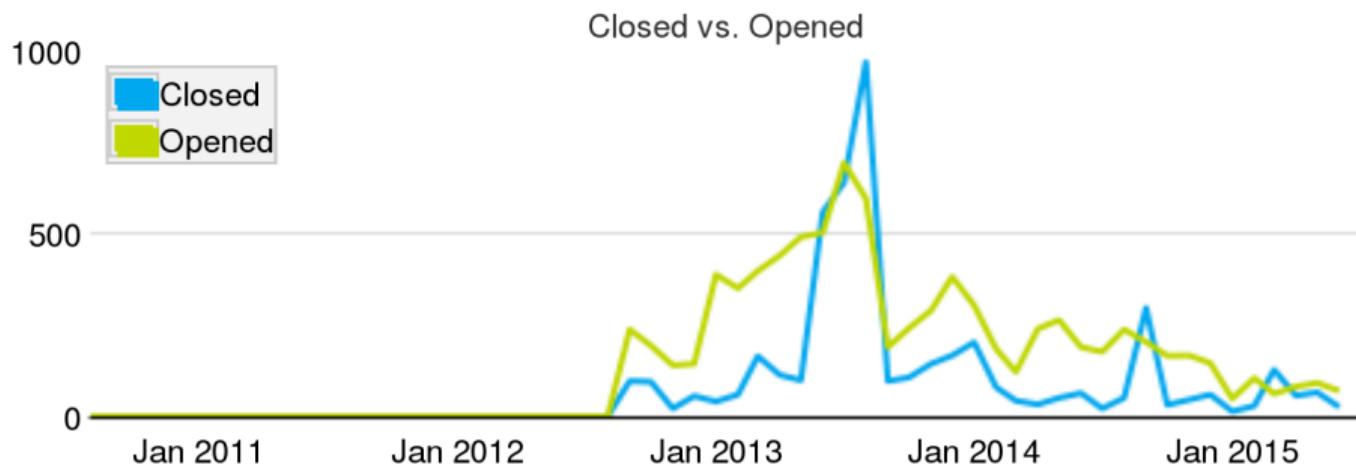
Performance

Demographics

Diversity

Final remarks

Tickets



Analytics with
GrimoireLabJesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

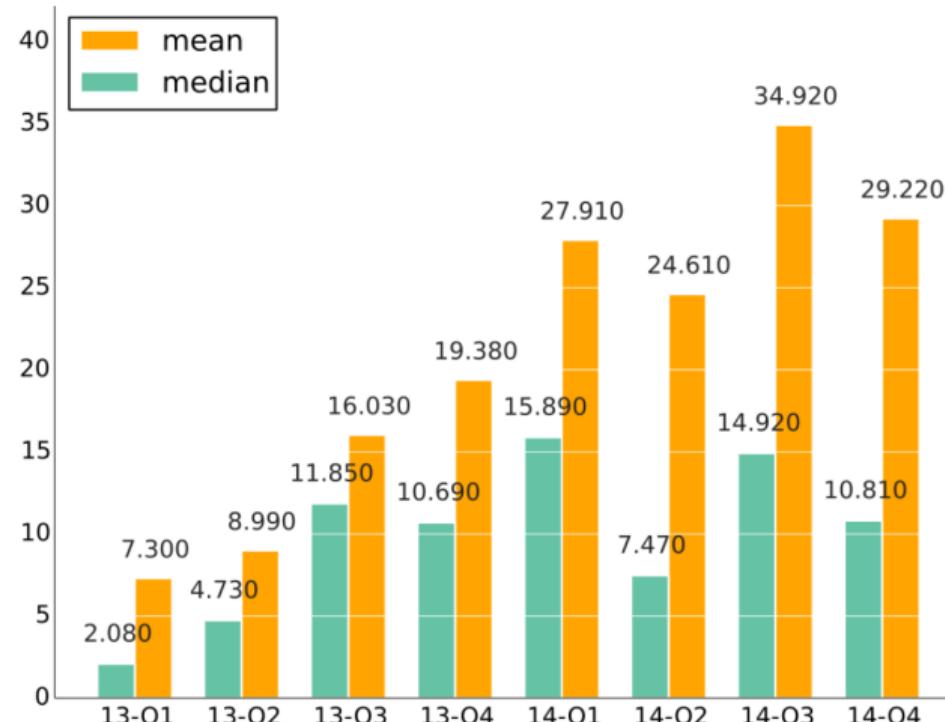
Performance

Demographics

Diversity

Final remarks

Review: time to merge



Analytics with
GrimoireLabJesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

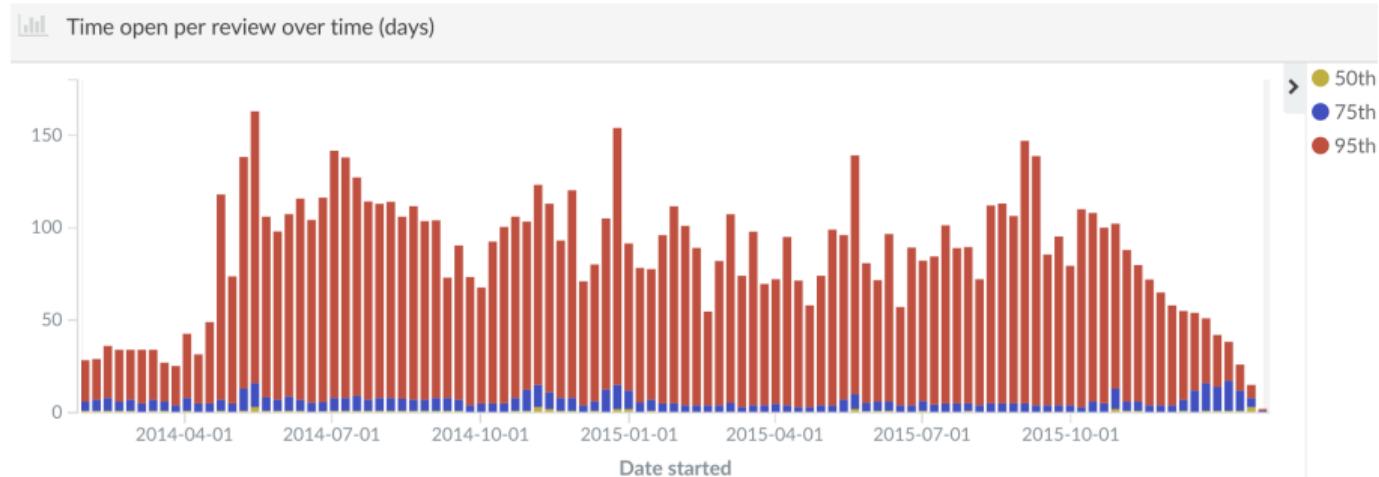
Performance

Demographics

Diversity

Final remarks

Review: time to merge



Analytics with
GrimoireLabJesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

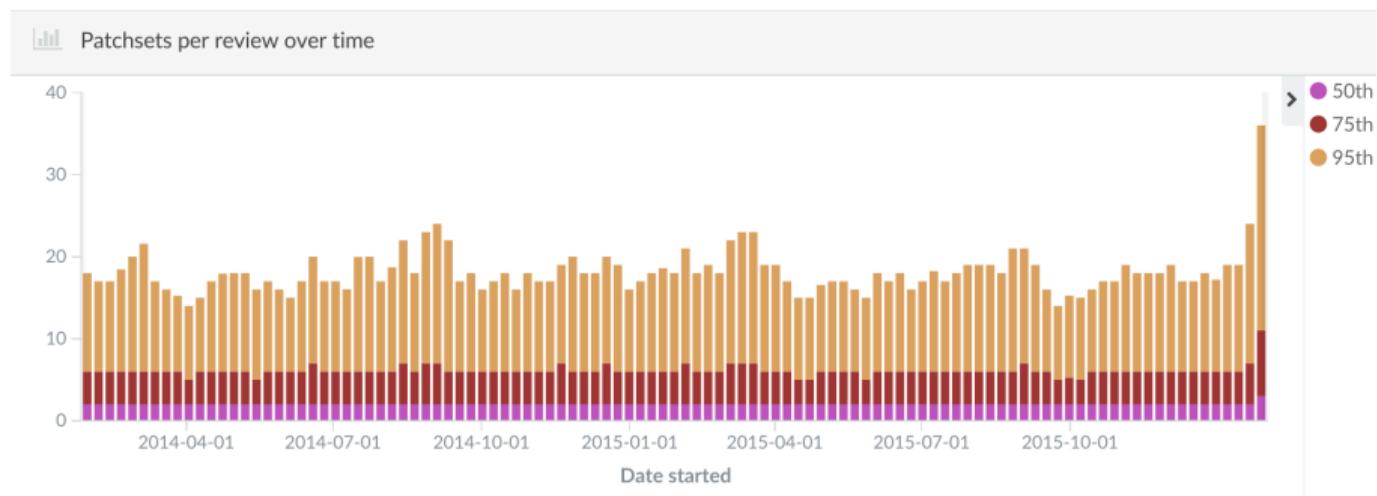
Performance

Demographics

Diversity

Final remarks

Versions per review



The coding process

From idea to implementation

- Story, design
- Ticket(s)
- Code review
- Automated testing
- Commit in code base

The OpenStack case

- Blueprint (if feature), Launchpad
- Ticket (bug, feature), Launchpad
- Code review, Gerrit
- Automated testing, Jenkins
- Commit in code base, Gerrit, Git

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

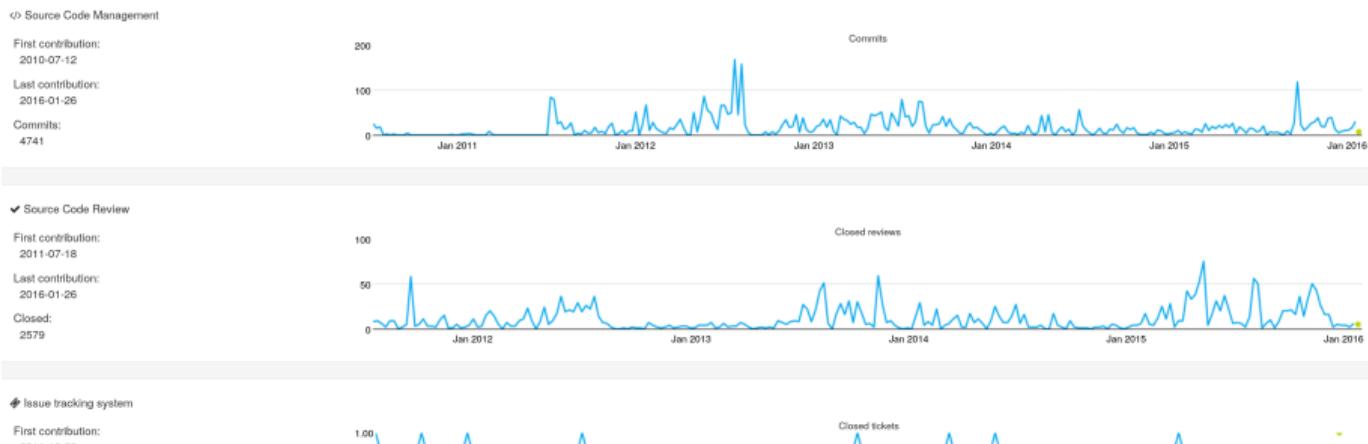
Performance

Demographics

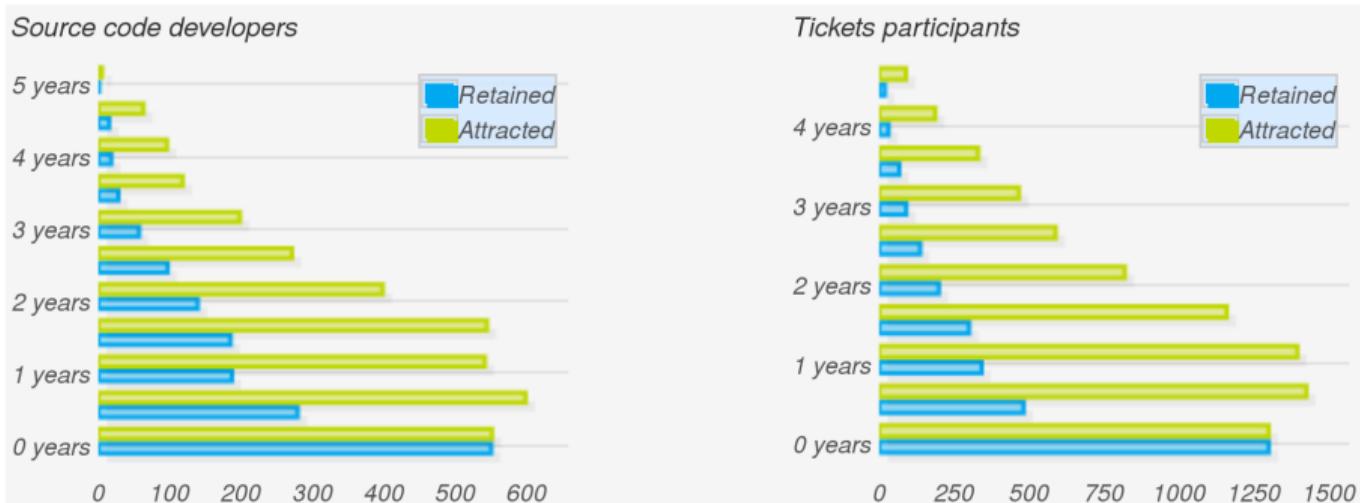
Diversity

Final remarks

- The repository level.
- The class of repository level.
- The project level.
- The global level.



The aging chart



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

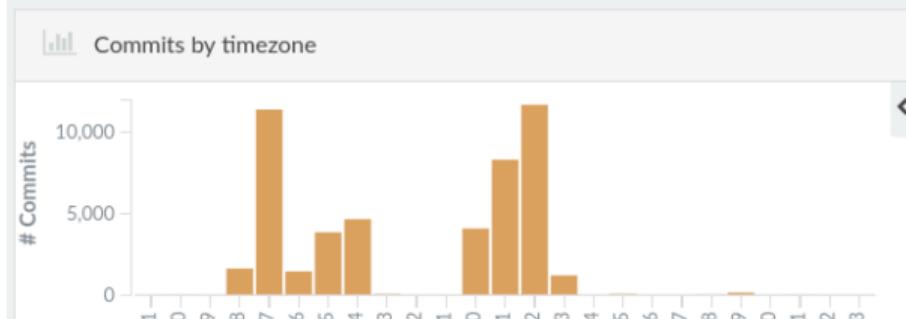
GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

Time zones



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

Demographics

Diversity

Final remarks

GitHub profiles



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

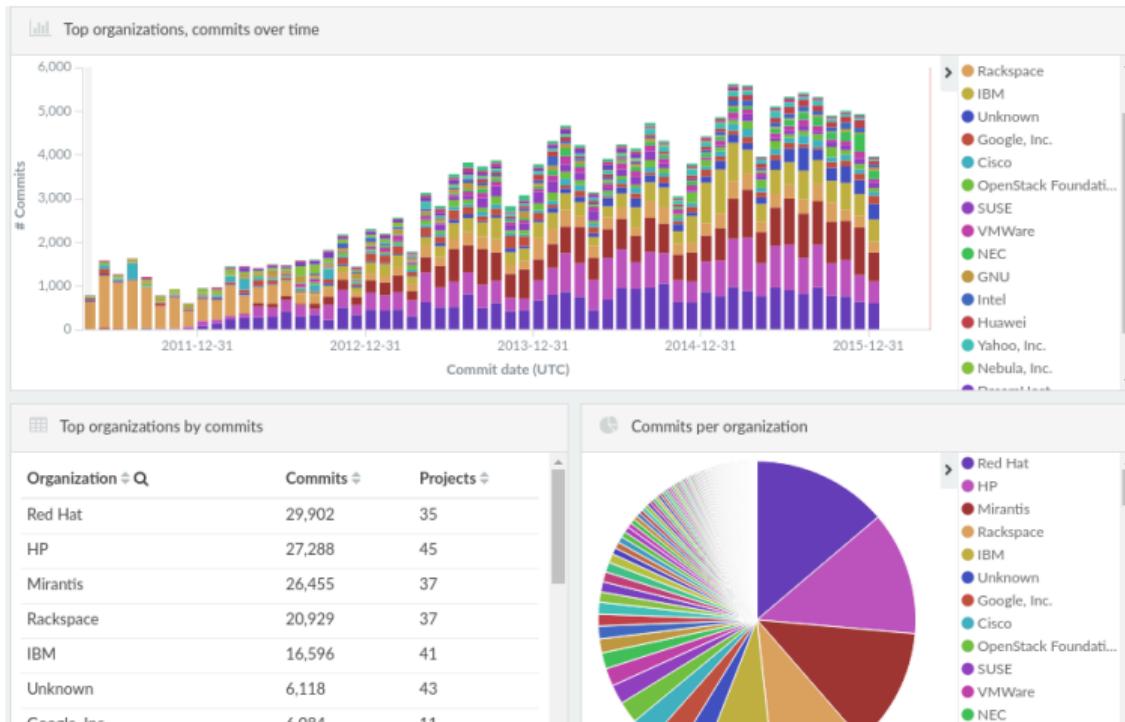
Performance

Demographics

Diversity

Final remarks

Affiliation



Apache Pony Factor

Pony Factor (PF) shows the diversity of a project in terms of the division of labor among committers in a project.

*Pony Factor is determined as: “**The lowest number of committers whose total contribution constitutes the majority of the codebase**”*

ke4qqq.wordpress.com/2015/02/08/pony-factor-math/

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks

Bitergia Elephant Factor



Bitergia Elephant Factor

The elephant factor shows the diversity of a project in terms of the division of labor among companies (by mean of developers affiliated with them).

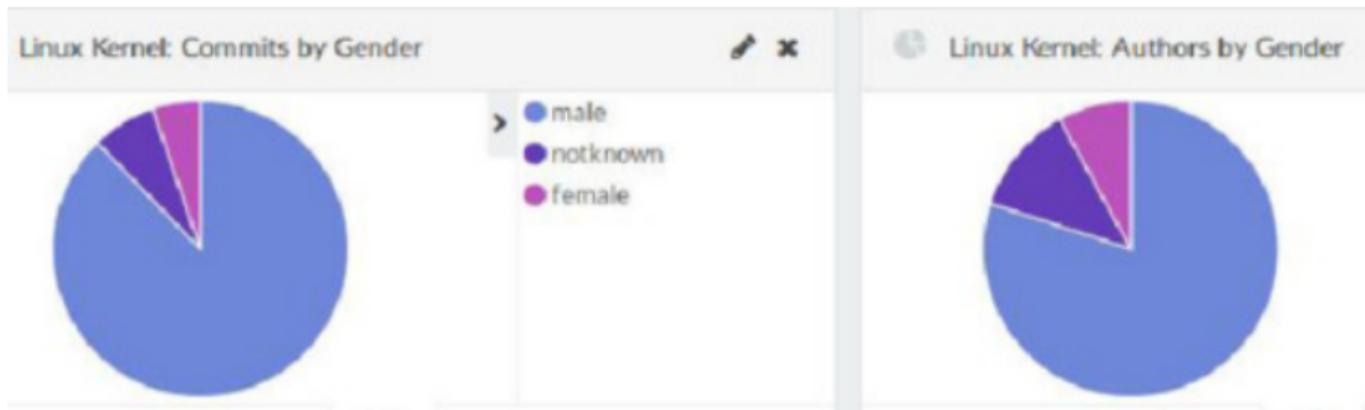
Elephant factor is determined as:

“The lowest number of companies whose total contribution (in commits by their employees) constitutes the majority of the commits”

Some projects (2016)

	Pony Factor	Elephant Factor	Commits (excl bots)
OpenNebula	4	1	12K
Eucalyptus	5	1	25K
CloudStack	14	1	42K
OpenStack	>100	6	126K
CloudFoundry	41	1	60K
OpenShift	10	1	15K
Docker	15	1	18K
Kubernetes	12	1	7K

Diversity: Gender gap



Commits by women: 6.8 % (4 Kcommits)

Women: 9.9 % (330 developers)

Linux kernel, Nov 2015 – Oct 2016

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks



Room for improvement

- Many other aspects... explore your own
- Refine what is important
- Explore new ways of making data useful
- Tell interesting stories based on data
- Visualization is very important
- Higher-order metrics
- Simplify results, make them meaningful

Summary

You cannot improve
what you cannot measure

Fortunately, you can measure a lot of things...

<http://chaoss.github.io/grimoirelab>

Credits (1)

- “Man With Two Hats”
Statue by Henk Visch, located in Ottawa, Canada
Picture by Lezumbalaberjenja in Wikimedia Commons
License: Public domain
https://commons.wikimedia.org/wiki/File:Man_With_Two_Hats_Ottawa_Statue_by_lezumbalaberjenja.jpg
- “Crowd at FOSDEM 2008”
by Jesús Corrius
License: CC Attribution 2.0
<http://www.flickr.com/photos/jcorrius/2302302707/>

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity

Final remarks



©2016-2019 Jesus M. Gonzalez-Barahona.

Some rights reserved. This document is distributed under the terms
of the Creative Commons License “Attribution-ShareAlike 4.0”,
available in

<http://creativecommons.org/licenses/by-sa/4.0/>

This document (including source) is available from
<https://github.com/jgbarah/presentations>