

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 in FOSS development

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 in FOSS development

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 in FOSS development

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 in FOSS development

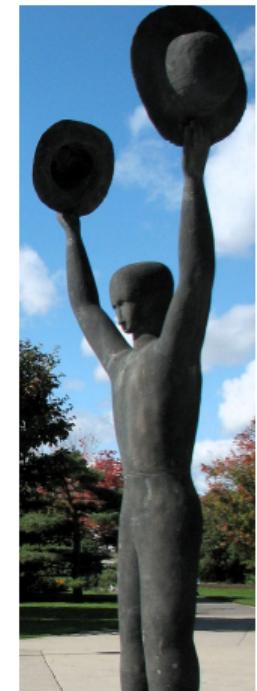
Final remarks

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 in FOSS development

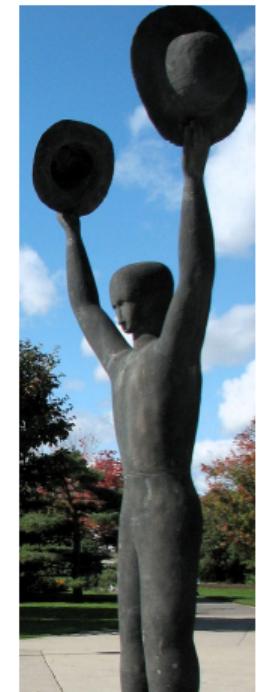
Final remarks

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 in FOSS development

Final remarks

Cauldron Alpha

The screenshot shows the Cauldron Alpha interface for the Django project. At the top, there's a navigation bar with a logo, a search bar containing 'Cauldron (a version)', a 'Send us feedback!' button, and a user profile icon. Below the navigation is a section titled 'Add data sources' with four options: GitHub, Git, GitLab, and Meetup. Each option has an input field for a URL and a 'Add' button. A note at the bottom of this section states: '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!'. The main content area is titled 'Django' and shows a table of 36 data sources. The table columns are 'Status', 'Data source', 'Last refresh', and 'Duration'. Each row contains a green checkmark, a GitHub icon followed by a URL, and timestamped logs, delete, and refresh buttons.

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

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

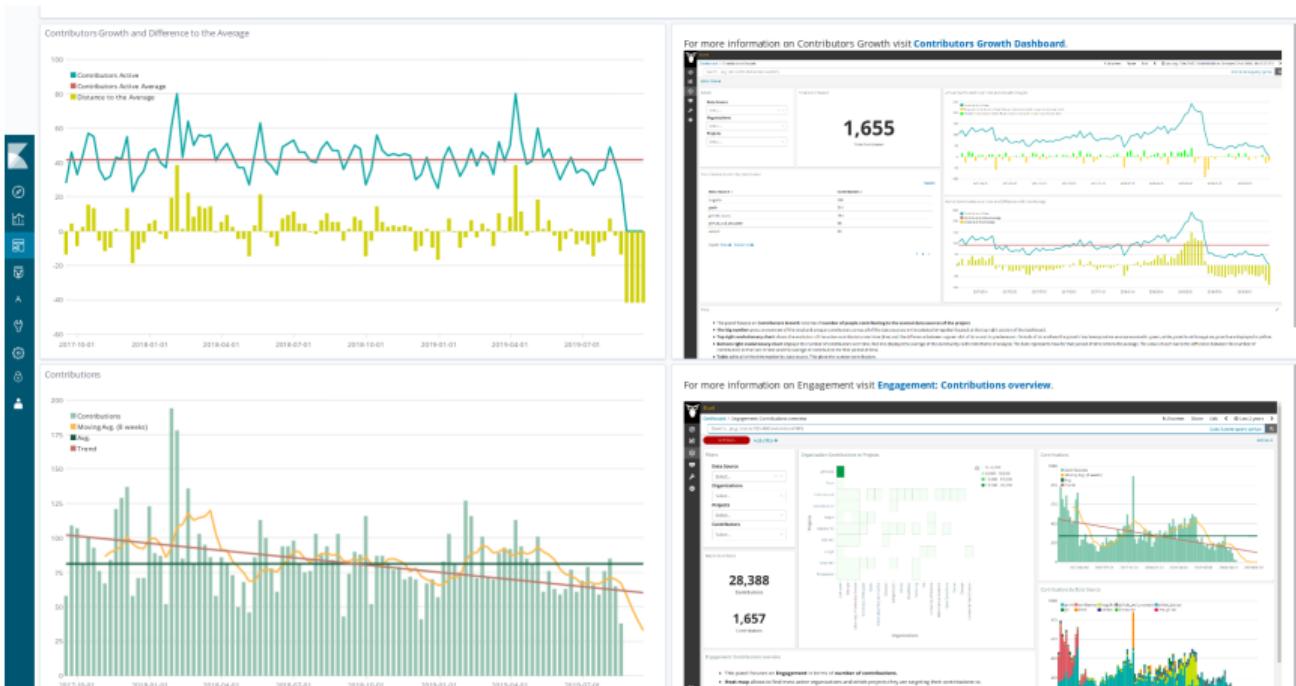
Performance

Demographics

Diversity in FOSS development

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 in FOSS development

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 in FOSS development

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 in FOSS development

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 in FOSS development

Final remarks



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

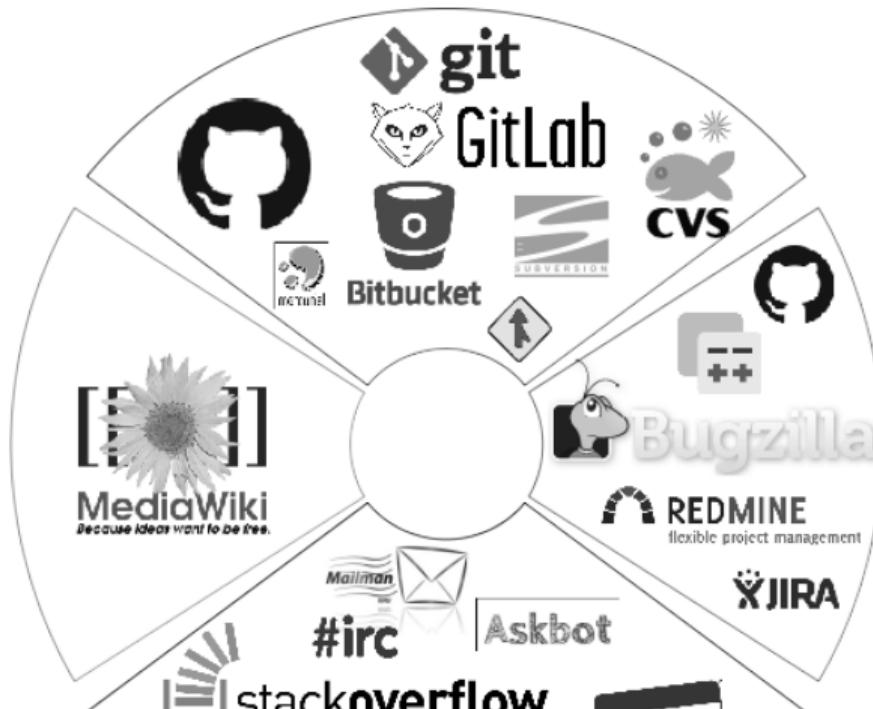
GrimoireLab

Case studies

Activity
Remaining code
Performance
Demographics
Diversity in FOSS development

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 in FOSS development

Final remarks



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

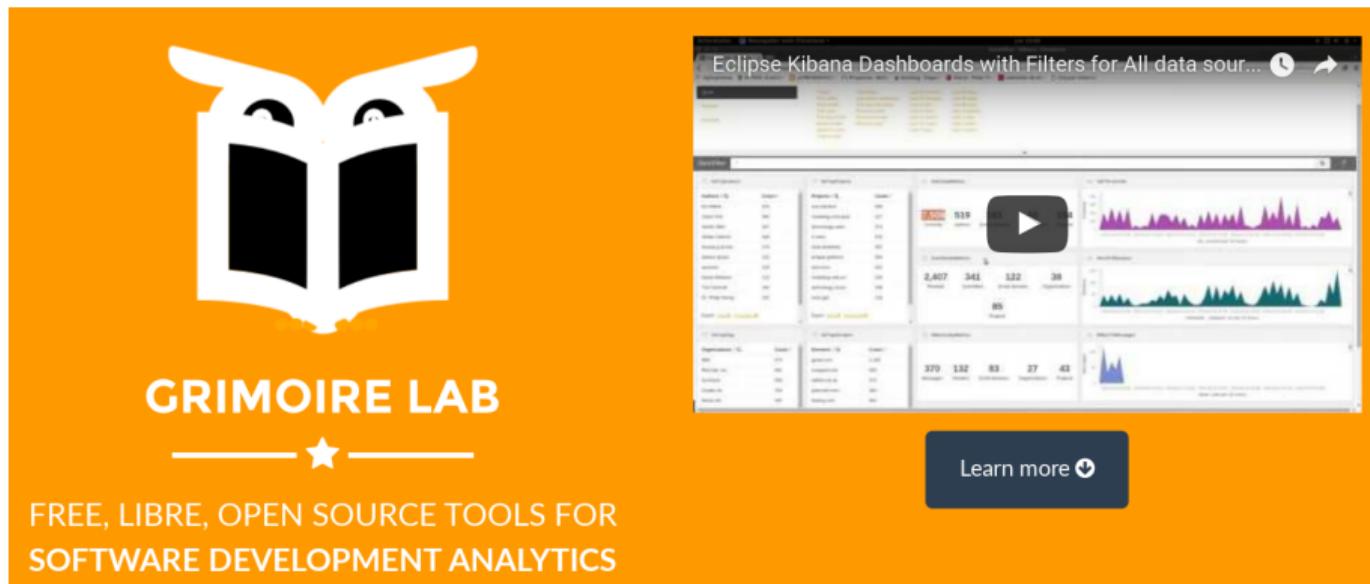
Remaining code

Performance

Demographics

Diversity in FOSS development

Final remarks



The image shows the GrimoireLab landing page. It features a large orange background with a white owl logo and the text "GRIMOIRE LAB". Below the logo is a star icon. The text "FREE, LIBRE, OPEN SOURCE TOOLS FOR SOFTWARE DEVELOPMENT ANALYTICS" is displayed. To the right, there is a screenshot of the Eclipse Kibana Dashboards interface, showing various data visualizations and filters. A "Learn more" button is also visible.

<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

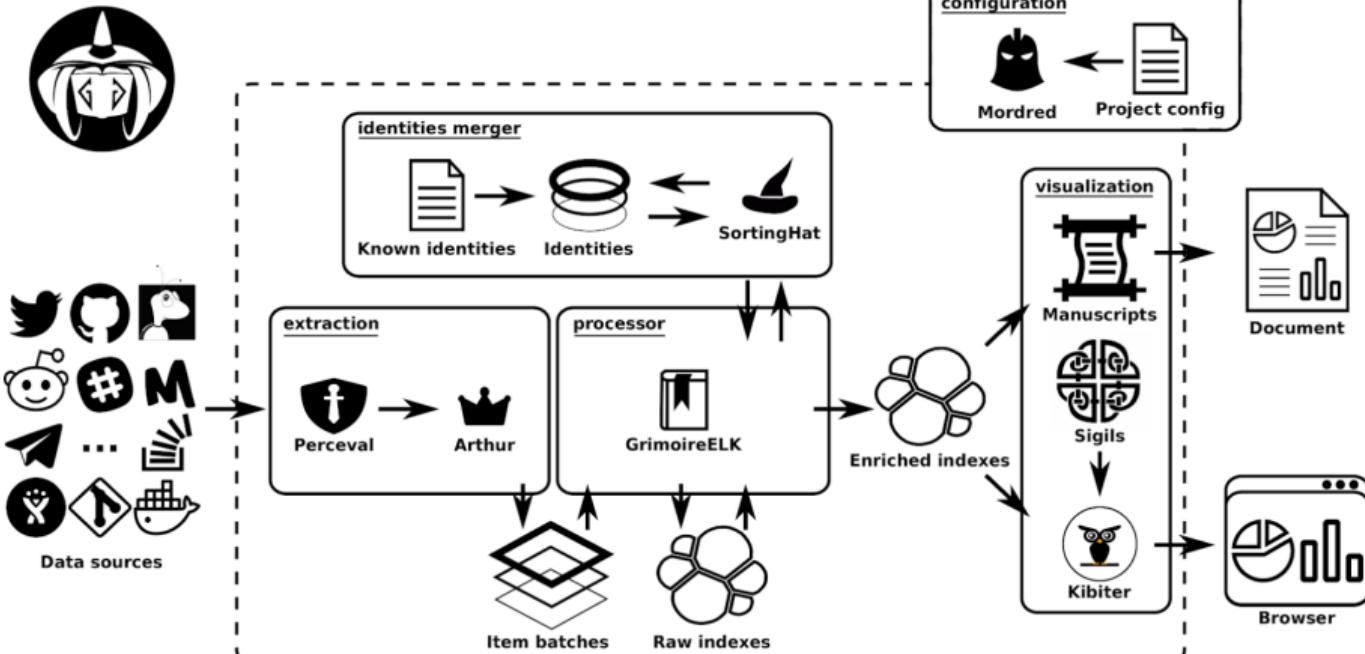
Remaining code

Performance

Demographics

Diversity in FOSS development

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 in FOSS development

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 in FOSS development

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 in a git repository")
parser.add_argument("repo", help = "Repository url")
parser.add_argument("--print", action='store_true', help = "Print the 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 in FOSS development

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 in FOSS development

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 in FOSS development

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

Activity / size (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

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 in FOSS development

Final remarks

Activity / size (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

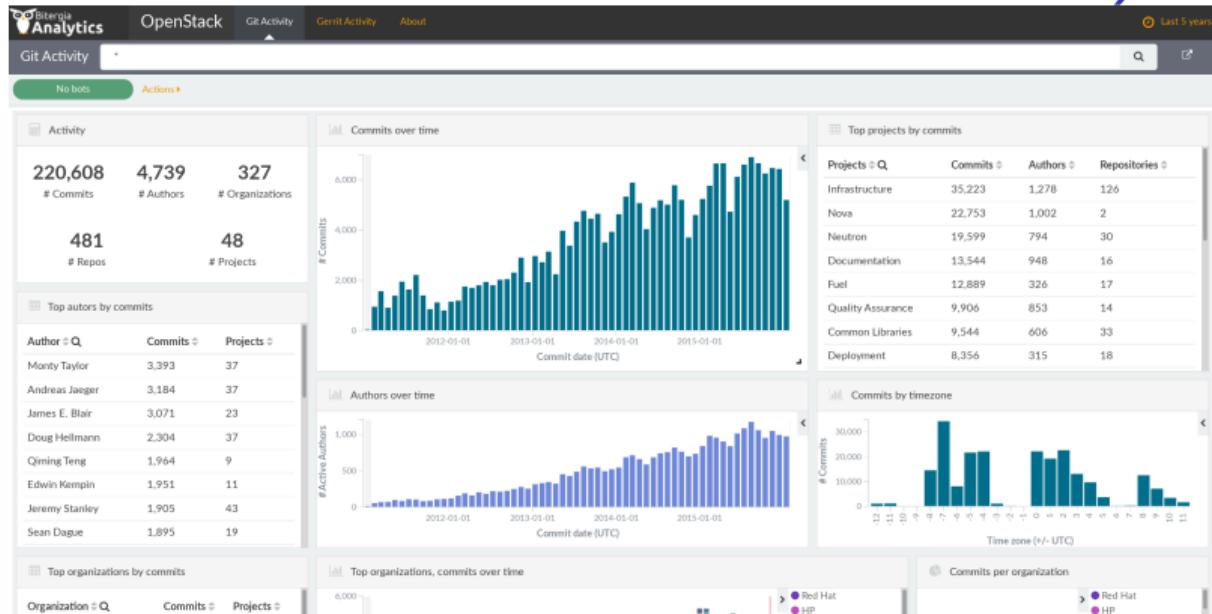
Performance

Demographics

Diversity in FOSS development

Final remarks

Activity / size (many facets)



Analytics with
GrimoireLabJesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

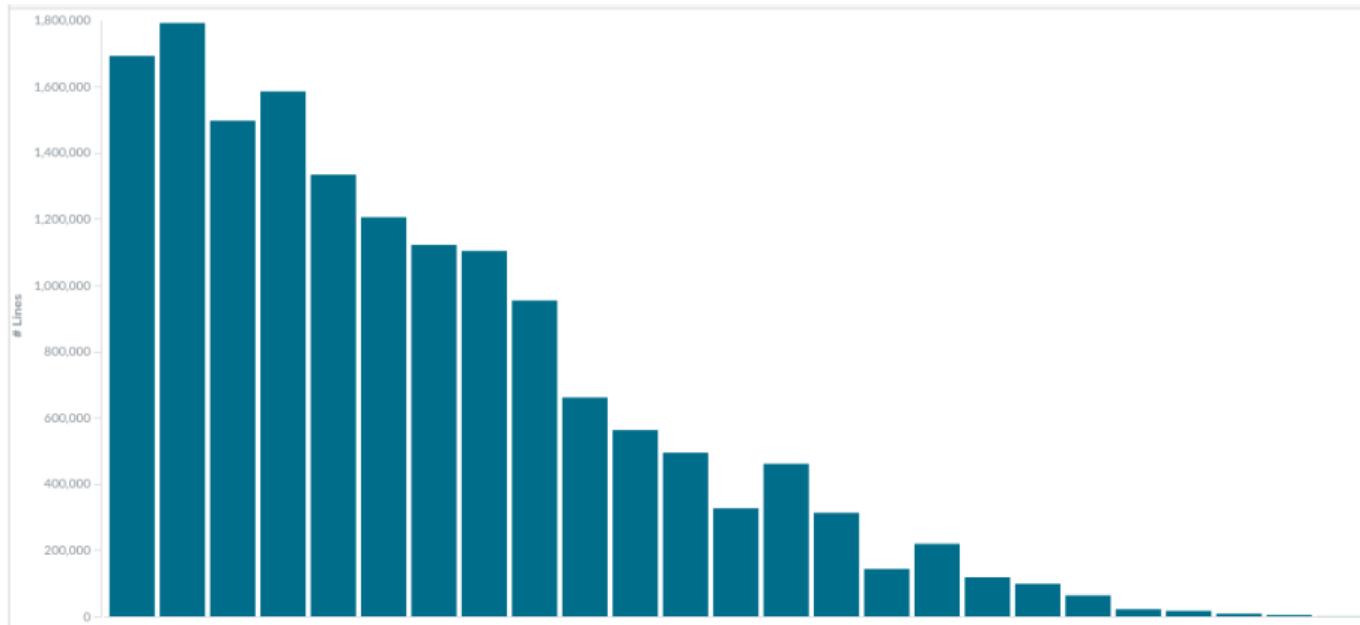
Performance

Demographics

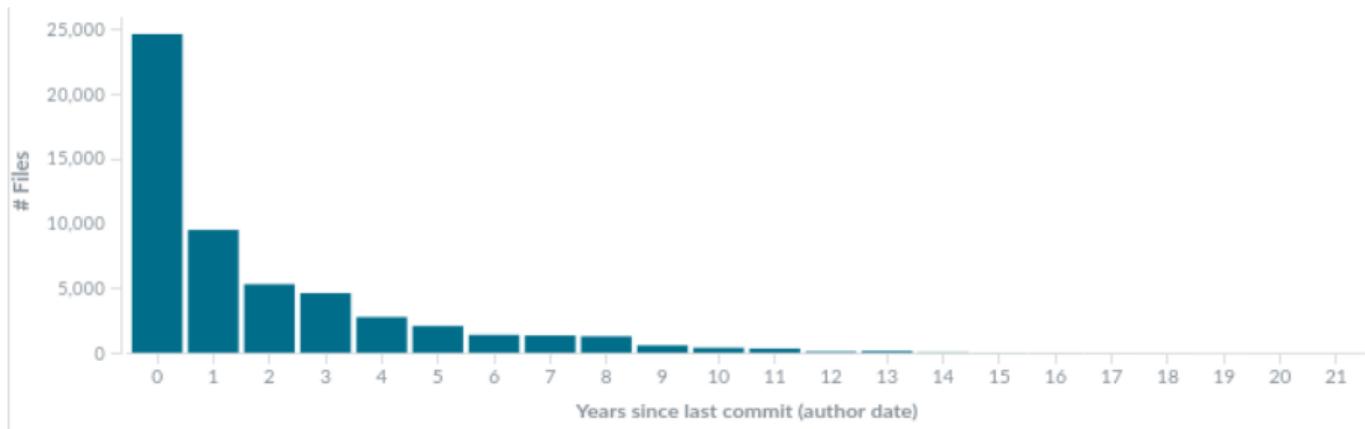
Diversity in FOSS development

Final remarks

How old is code

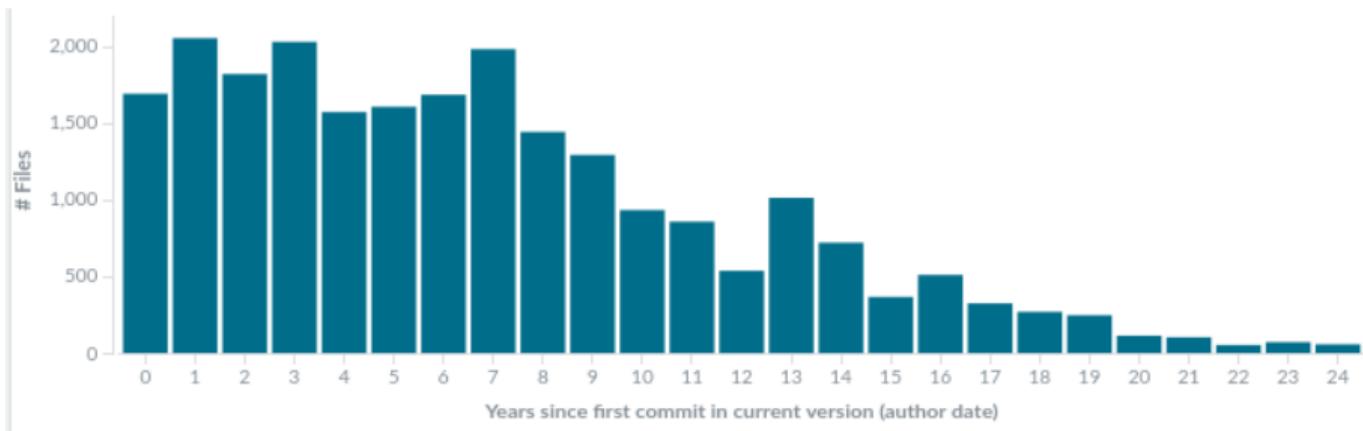


How old is code (2)



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

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

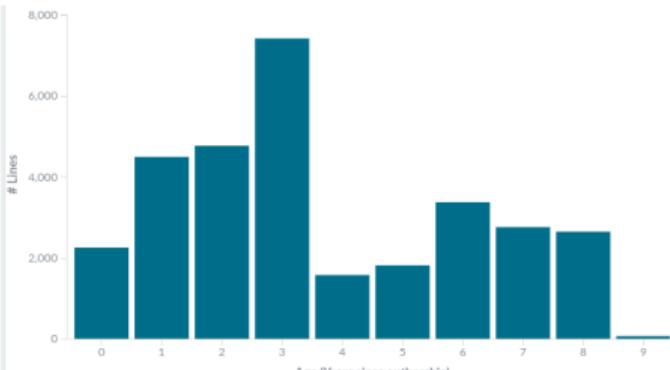
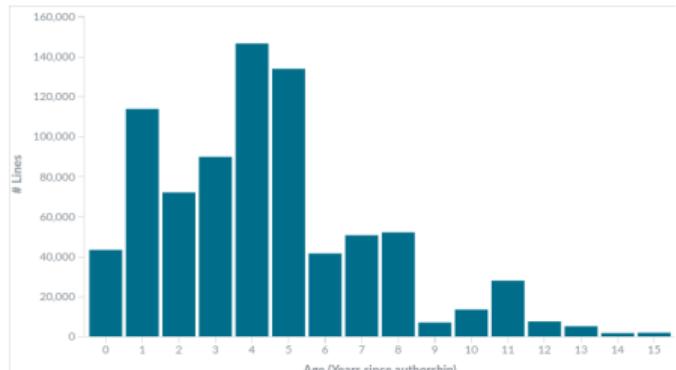
Performance

Demographics

Diversity in FOSS development

Final remarks

How old is code? drivers/net in Linux



Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

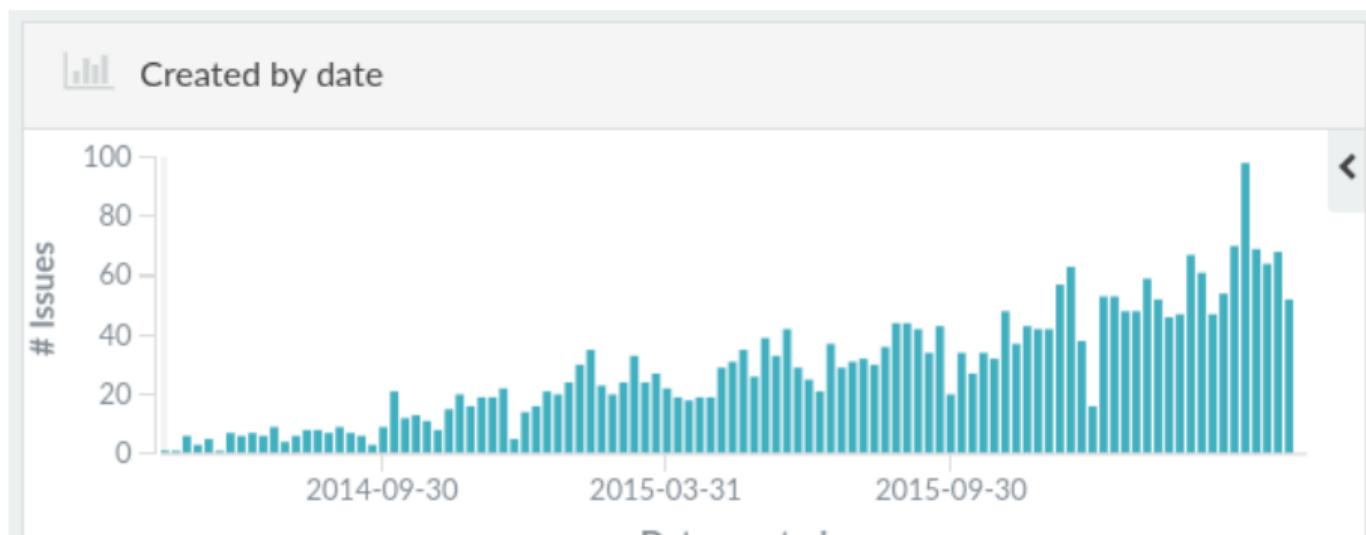
Performance

Demographics

Diversity in FOSS development

Final remarks

Backlog (evolution over time)



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

Performance

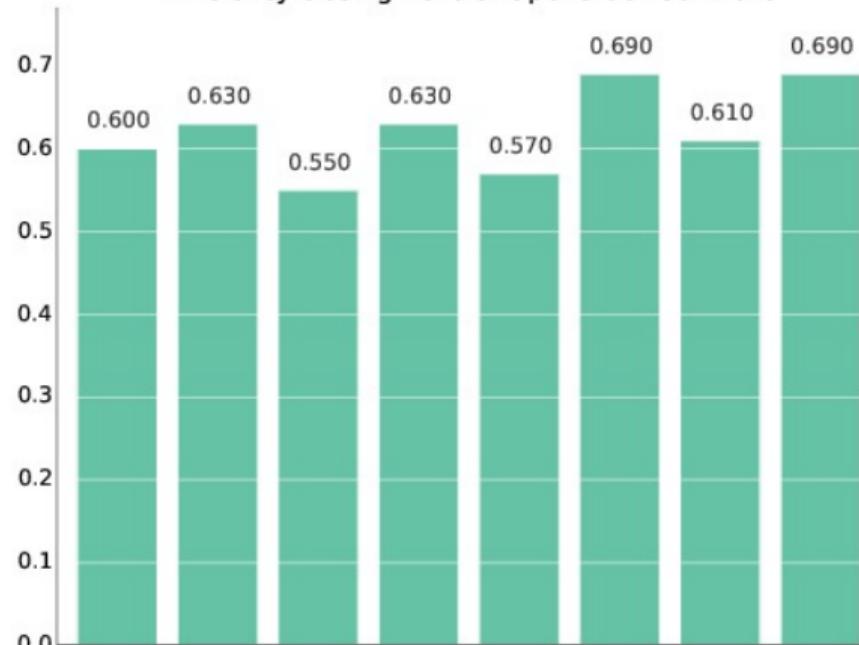
Demographics

Diversity in FOSS development

Final remarks

Efficiency

Efficiency closing tickets: OpenStack Software



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

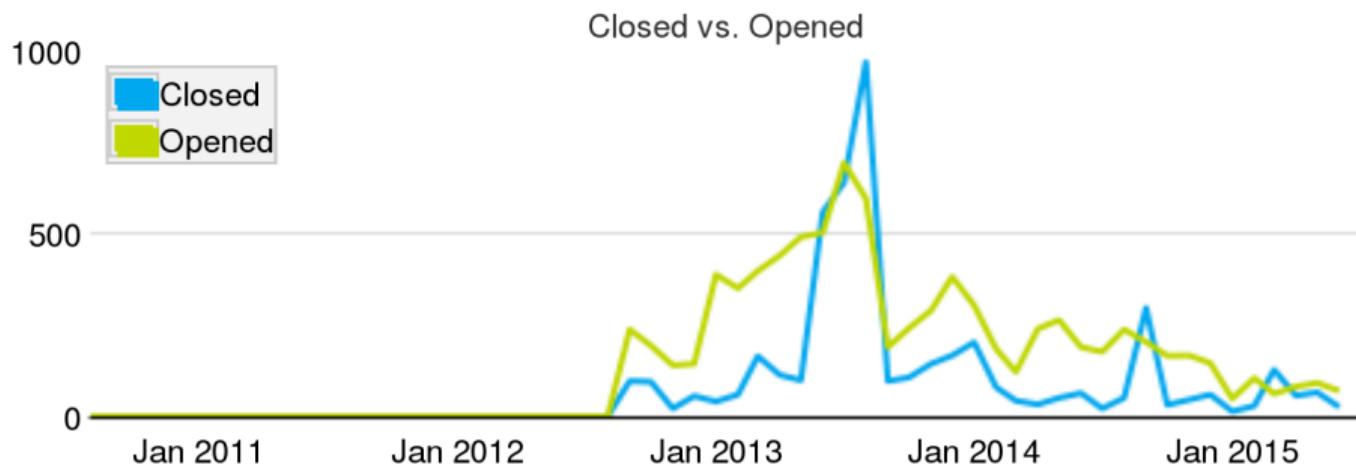
Performance

Demographics

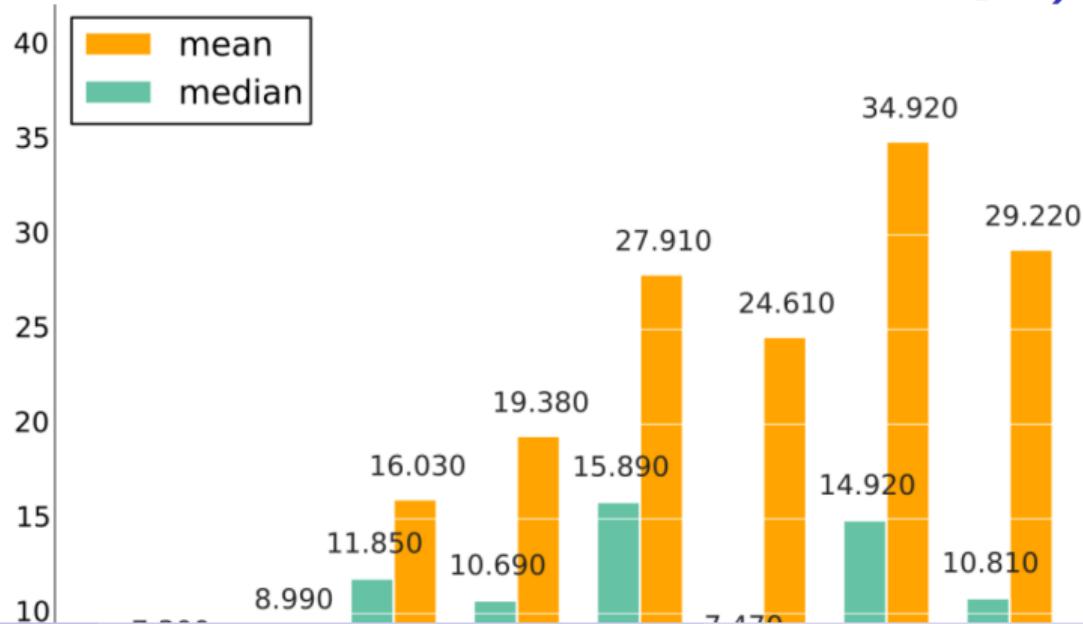
Diversity in FOSS development

Final remarks

Tickets



Code review (time to merge)



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

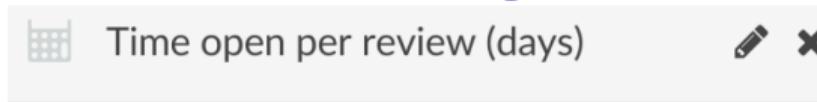
Performance

Demographics

Diversity in FOSS development

Final remarks

Code review (time to merge, metrics)



0.33

50th percentile of timeopen

5

75th percentile of timeopen

67.45

95th percentile of timeopen

Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

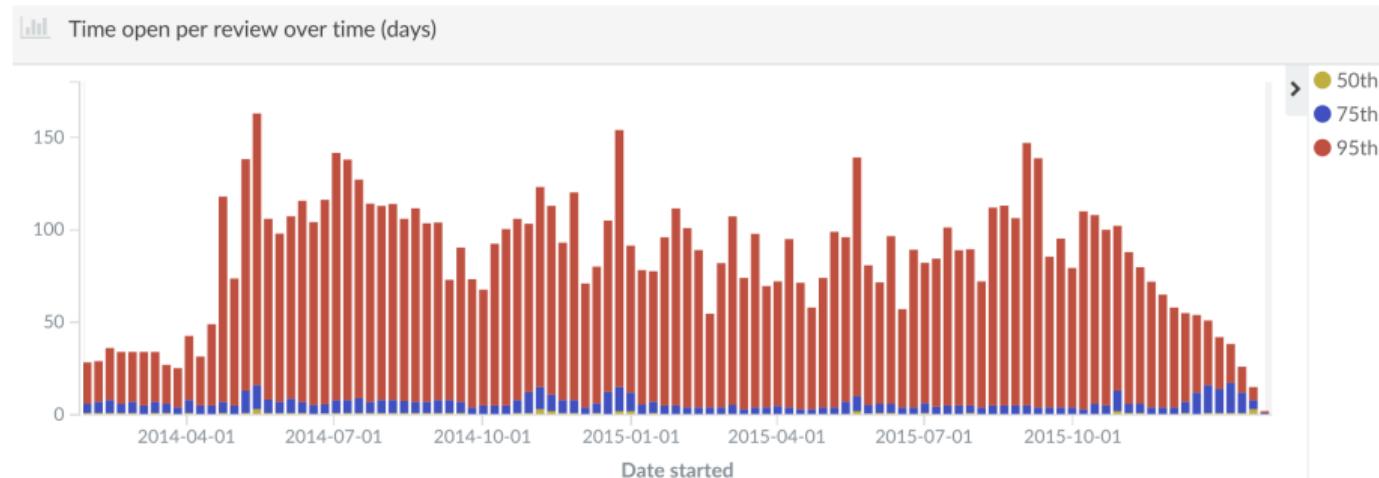
Performance

Demographics

Diversity in FOSS development

Final remarks

Code review (time to merge, evolution)



Analytics with GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

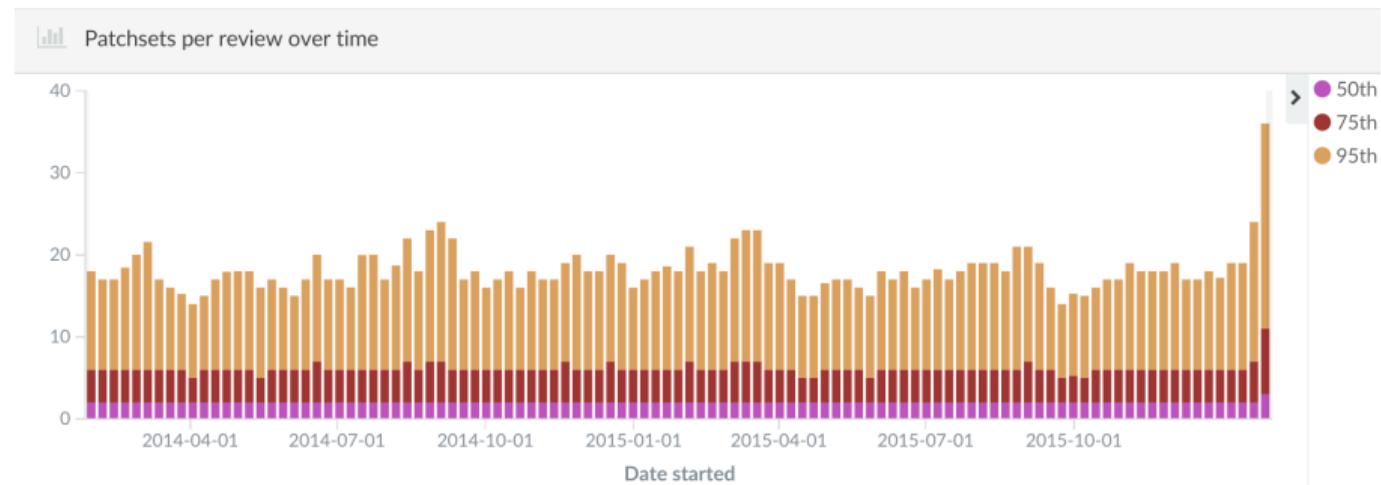
Performance

Demographics

Diversity in FOSS development

Final remarks

Code review (number of versions per review)



The complete coding process

From idea to implementation

- Story, design
- Ticket(s)
- Code review
- Automated

The OpenStack case

- Blueprint (if feature), Launchpad
- Ticket (bug, feature), Launchpad
- Code review, Gerrit

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 in FOSS development

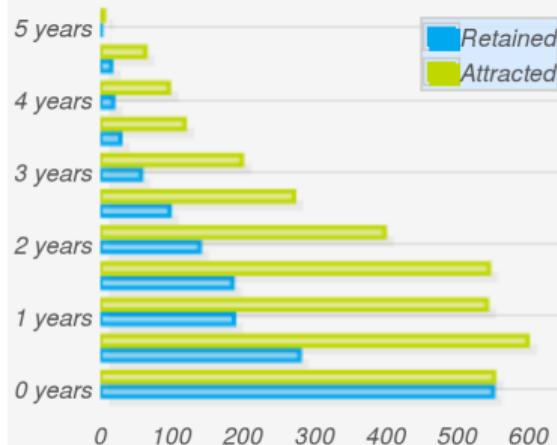
Final remarks

The many identities of anyone

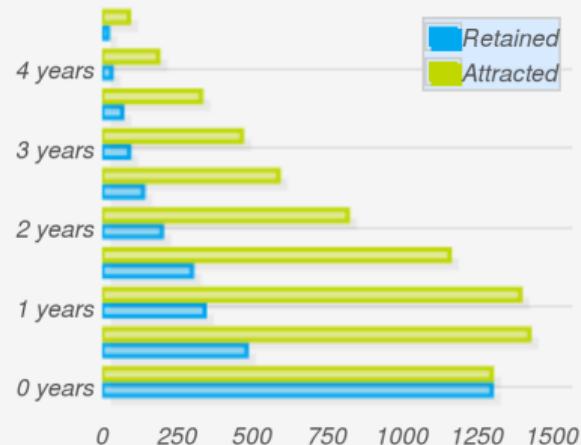
- The repository level.
- The class of repository level.
- The project level.
- The global level.

Demographics: The aging chart

Source code developers



Tickets participants



Communities of volunteers

- “Peripheral”: activities (questions, reporting bugs)
- Small

Demographics: Contributors funnel

Inner source

- Questions, reports, etc. in public (no more coffee machine meetings)
- Moving to develop: answers, bug fixes

Demographics: Mentorship

- Helping newcomers, helping people from other areas
- Usually linked to bug fixing and code review
- Who is helping others to improve their skills?

Analytics with
GrimoireLab

Jesus M.
Gonzalez-Barahona

A bit of context

Dealing with
dynamic
complexity

Data sources

GrimoireLab

Case studies

Activity

Remaining code

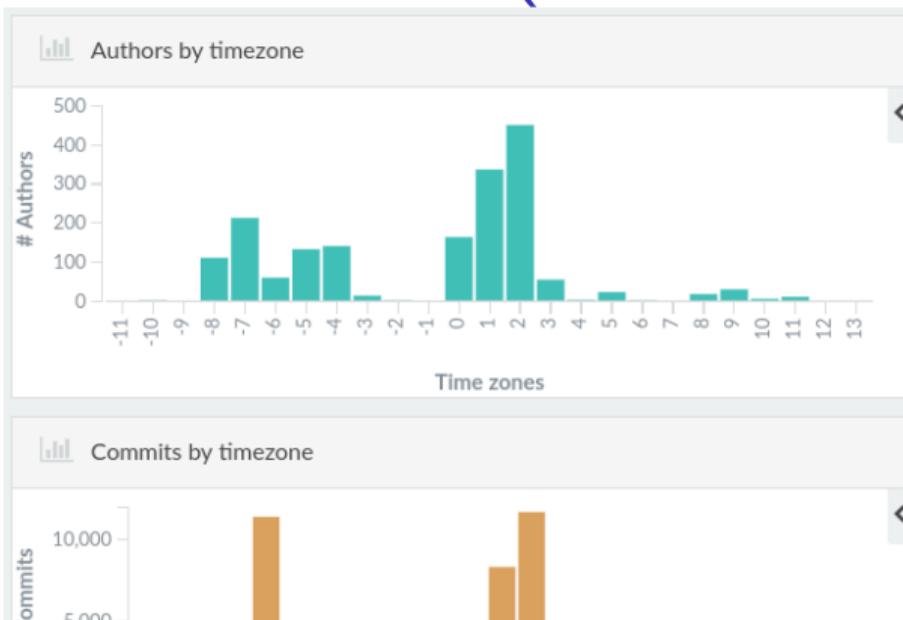
Performance

Demographics

Diversity in FOSS development

Final remarks

Diversity: geographical information (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 in FOSS development

Final remarks

Diversity: geographical information (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

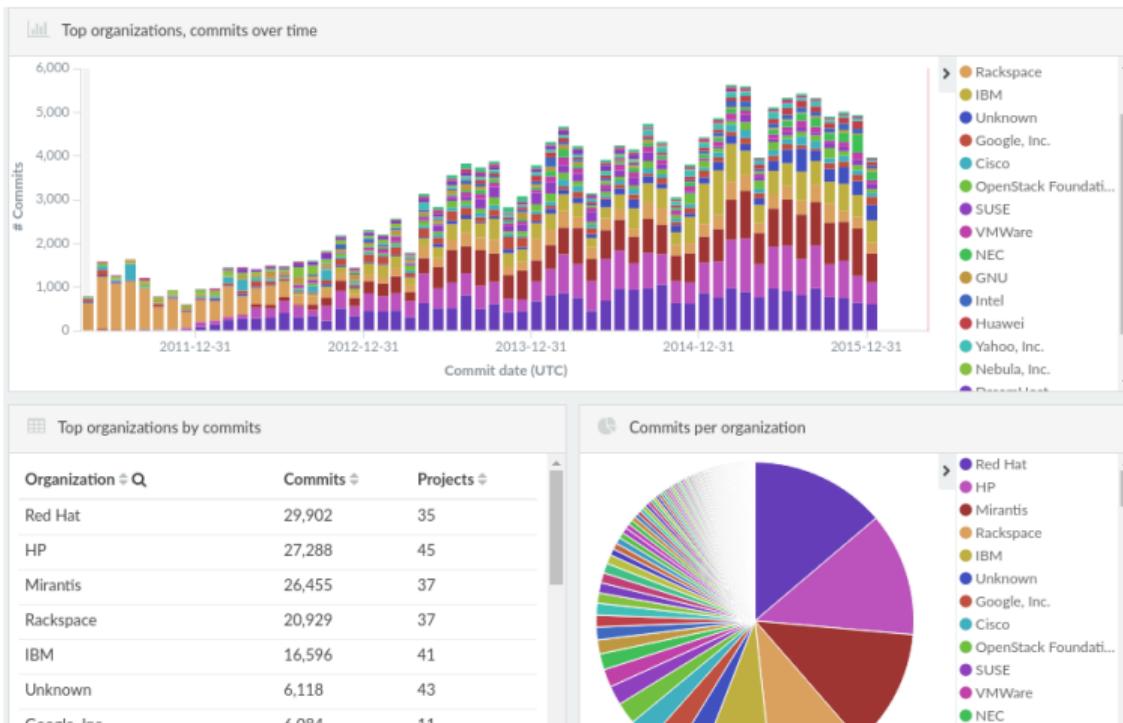
Performance

Demographics

Diversity in FOSS development

Final remarks

Diversity: affiliation



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 in FOSS development

Final remarks

Diversity: Apache Pony Factor

In words of Daniel Gruno:

We [the ASF] created a term we have coined “Pony Factor” (because ASF is full of ponies, or people who think they are ponies). Pony

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 in FOSS development

Final remarks

Diversity: Bitergia Elephant Factor



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 in FOSS development

Final remarks

Diversity: Bitergia
Elephant Factor
Projects can benefit from powerful collaborations from companies (elephants). The elephant factor shows the diversity of a project in terms of the

Diversity: some projects

	Pony Factor	Elephant Factor	Com
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	10	1	7K

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 in FOSS development

Final remarks

Diversity: Code “owned”



*“The land
belongs
to its workers”*

Emiliano Zapata

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 in FOSS development

Final remarks

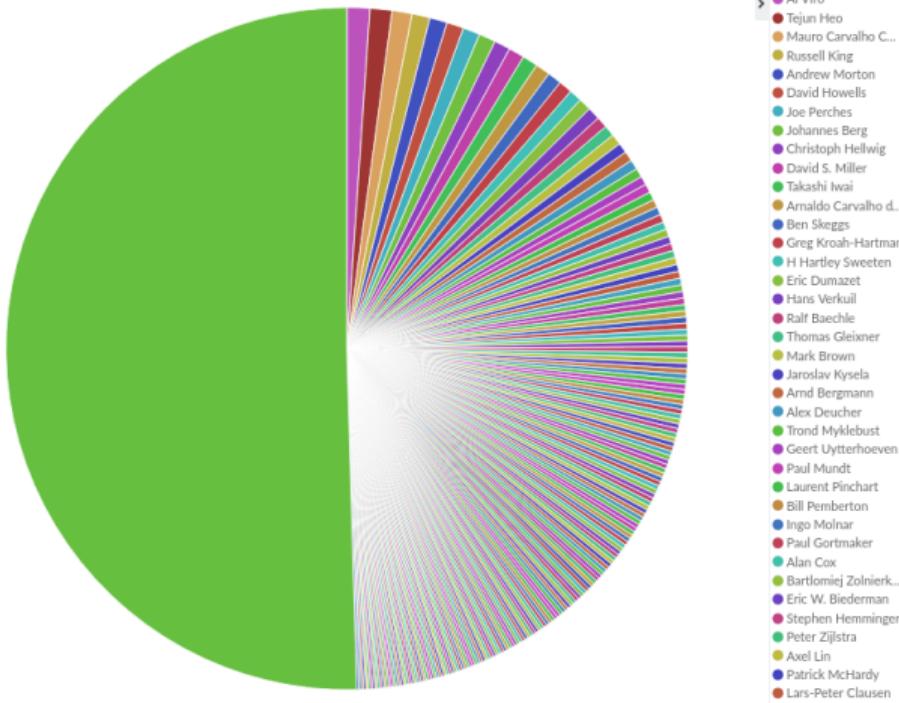
Diversity: Code “owned”

The code changes over time. The current version is “owned” by the people who produced it.

The code “belongs” to those who wrote it.

Zapata factor (*work in pro-*

Diversity: Code “owned”



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 in FOSS development

Final remarks

Diversity: Code “owned”
The code “belongs” to companies who employ developers changing it.

United Fruit factor (work in progress):

“The lowest number of

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 in FOSS development

Final remarks

Diversity: Gender gap



Commits by women: 6.8 % (4 Kcommits)

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 in FOSS development

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

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 in FOSS development

Final remarks

Summary

You cannot
improve
what you cannot

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:](https://commons.wikimedia.org/wiki/File:Man_With_Two_Hats_Ottawa_Statue_by_lezumbalaberjenja.jpg)

Man_With_Two_Hats_Ottawa_Statue_by_

lezumbalaberjenja.jpg

- “Crowd at FOSDEM 2008”

by Jesús Corrius

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 in FOSS development

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>