Introducing CHAOSS: A Community for Advancing Project Transparency

Open Source Summit NA 2017
Sept. 11th 2017, Los Angeles, CA, USA

Matt Germonprez, University of Nebraska at Omaha
Jesus M. Gonzalez-Barahona, Bitergia / URJC
Harish Pillay, Red Hat
Sean Goggins, University of Missouri
Why?

We need to understand how software is being developed:

• Open Development Analytics
• Project Health

Activity, community, processes, signals, values, and goals.
Why?
Open Development Analytics
Why? Open Development Analytics

How are we writing our code?

Understanding developing processes based on facts
Why? Open Development Analytics

Analytics can help to understand
Why?

Open Development Analytics

A step beyond in project transparency

From “show me the code” to “Show me the numbers”
Why?
Project Health

• No singular health determination can be made across all open source projects, however:
  • We can start to understand what composite metrics signal and how they can be related to actions

• We aim to provide insight as local interpretations are done on the metrics -- provide guideposts for what others have done in similar contexts and explicate how peer communities compare
Why? Use Case

Software Health Use Case: Should I Engage in this Project?

The Use Case
Visual Representations
Metrics Standard Implementation

Metric 1: Downloads
Metric 2: Developers
Data Endpoint
ghData
Corporate Repository

Metrics Standard Data Mapping Metric Source Data
What?

chaoss.community
**Mission**

Produce integrated, open source software for analyzing software development, and definition of standards and models used in that software in specific use cases.

Establish implementation-agnostic metrics for measuring community activity, contributions, and health.

Optionally produce standardized metric exchange formats, detailed use cases, models, or recommendations to analyze specific issues in the industry/OSS world.
How?

“Meeting point for people defining & using software metrics and people implementing analytics for software development”
Founding Members

THE LINUX FOUNDATION
mozilla
eclipse
SYMPHONY SOFTWARE FOUNDATION

redhat
Bitergia
JONO BACON CONSULTING
openstack

UNIVERSITY OF Nebraska Omaha
University of Missouri
University of Victoria
SECO
Software Sustainability Institute

CLOUD NATIVE COMPUTING FOUNDATION
SAUCE LABS
Linaro
How?
Structure

Two Committees:

**Metrics**
Implementation-agnostic community development metrics

**Software**
Integrated FOSS tools for software development analytics
How?
Metrics Committee

Community Developing Metrics

<table>
<thead>
<tr>
<th>Diversity</th>
<th>Ecosystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk</td>
<td>Growth-Maturity-Decline</td>
</tr>
</tbody>
</table>

Metric → Signal → Actions

wiki.linuxfoundation.org/oss-health-metrics
2) Project Growth - Maturity - Decline

An OSS community goes has states: growth, maturity, and decline. The state that a community is in may prove important when evaluating both across and within community concerns.

2) Project Growth - Maturity - Decline

GHTorrent: Time between opening and a committer responding to an issue (single project)

```
SELECT issues.id AS "issue_id",
       issues.created_at AS "created_at",
       MIN(issue_comments.created_at) AS "responded_to"
FROM issues
JOIN issue_comments
ON issue_comments.issue_id = issues.id
WHERE issue_comments.user_id IN
    (SELECT users.id
     FROM users
     JOIN commits
     WHERE commits.author_id = users.id
     AND commits.project_id = 78852)
AND issues.repo_id = 78852
GROUP BY issues.id
```
Prototyping Human Centered Metrics: Enable Comparisons, Make Trends Central to the Use Experience

1. Comparisons
   - Z-score trailing average
   - 100% is the compared project

2. Ecosystem
   - Downloads / Day
   - rails/rails

3. Activity
   - akka/akka versus rails/rails
   - Commits / Week
   - Forks / Week
   - Issues / Week

4. Top Dependents
   - rspec-rails
   - haml
   - sidekiq
   - spree_core
   - simple_form
   - jquery-rails
   - factory_girl_rails
   - kaminari
   - carmenwave
   - shoulda

5. Top Dependencies
   - websocket-driver
   - node
   - rails-dom-testing
   - mail
   - rails-dom-testing
   - rails-html-sanitizer
   - rack-test
   - rack
   - rails-dom-testing
   - rails-html-sanitizer

Metric → Signal → Actions
How?
Software Committee

Prospector

credit

GRIMOIRE LAB
How? Software Committee Prospector - Why?

- Open source projects vary greatly in strength, significance, vibrancy and influence
- No simple way to evaluate or compare projects objectively, other than through individual experts
- Risk of committing to declining projects or missing out on thriving ones
- Open Source projects are not always openly trackable
How? Software Committee Prospector - How?

- Providing an **objective, consistent and repeatable** set of metrics of projects for **success, sustainability and vibrancy**.
- These can then coherently help assess and **track continuously** open source projects, which in turn would help drive the evolution of projects.
Metric datapoints:

- **Percentage of committers by dominant domain name**: 37.01%
  
  Rationale: If more than 50% of COMMITTERS are from one domain (via email ID) it is dominated by one set of people. Suggested target is to have it less than 35%.

- **Percentage of commits by dominant domain name**: 57.34%
  
  Rule: If more than 50% of CODE COMMITS are from one domain (via email ID) it is dominated by one set of people. Suggested target is to have it less than 35%.

- **Unique email address domains**: 345 domains
  
  How many unique domains are represented?

- **Unique committers**: 846 committers
  
  How many UNIQUE COMMITTERS are represented?

Number of commits over last:

- **3 months**: 850 commits from 81 contributors
- **6 months**: 1,985 commits from 131 contributors
- **9 months**: 2,984 commits from 164 contributors
- **12 months**: 4,176 commits from 233 contributors
- **24 months**: 6,028 commits from 412 contributors
How? Software Committee cregit

Framework to create evolutionary views of source code stored in a git repository, Allow the summarizing of contributions at token, function, or file level.

Current support for C, C++, Java, go, and python

github.com/cregit
How? Software Committee

**Cregit**: improving accuracy of git-blame

- Git-blame tracks changed lines, not tokens
  - Last person who modified part of a line, becomes “contributor” of the entire line
  - Cregit is capable of tracking the contributor of each token in a line

- In Linux:
  - blame per line is accurate in 75%
  - blame per token (using cregit) is accurate 95%
  - Results based on statistical sampling and manual analysis, with 95% reliability with +/-5% of error

- Currently in use by the Linux Kernel

[cregit.linuxsources.org](cregit.linuxsources.org)
How? Software Committee
cregit

cpu_notifier_register_begin();

if (create_hash_tables()) {
    err = -ENOMEM;
    goto out;
}

entry = proc_create("profile", S_IWUSR | S_IRUGO,
                    NULL, &proc_profile_operations);

if (!entry)
    goto out;
proc_set_size(entry, (1 + prof_len) * sizeof(atomic_t));
__hotcpu_notifier(profile_cpu_callback, 0);

out:
    cpu_notifier_register_done();
    return err;
}

Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Username</th>
<th>Contributions</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>William Lee Irwin III</td>
<td>william lee irwin iii</td>
<td>58</td>
<td>58.00%</td>
</tr>
<tr>
<td>Srivatsa S. Bhat</td>
<td>srivatsa s. bhat</td>
<td>26</td>
<td>26.00%</td>
</tr>
<tr>
<td>Paolo Ciarrocchi</td>
<td>paolo ciarrocchi</td>
<td>5</td>
<td>5.00%</td>
</tr>
<tr>
<td>David Howells</td>
<td>david howells</td>
<td>4</td>
<td>4.00%</td>
</tr>
<tr>
<td>Denis V. Lunev</td>
<td>denis v. lunev</td>
<td>4</td>
<td>4.00%</td>
</tr>
<tr>
<td>Al Viro</td>
<td>al viro</td>
<td>2</td>
<td>2.00%</td>
</tr>
<tr>
<td>Dave Hansen</td>
<td>dave hansen</td>
<td>1</td>
<td>1.00%</td>
</tr>
</tbody>
</table>
How? Software Committee

GrimoireLab

Software Development Analytics Toolset

- Retrieval from +30 data sources
- Storage of all metadata (ElasticSearch)
- Computing of interesting metrics
- Visualization
- Reports

grimoirelab.github.io
How? Software Committee

GrimoireLab

**PERCEVAL**
Sir Perceval goes on the quest to retrieve and gather data from git, GitHub, Bugzilla, JIRA, Gerrit, mbox, pipermail, StackExchange, Discourse, etc.

**SORTING HAT**
Tool to manage people identities information, to merge multiple person identities across different data sources, enrich profile information with affiliation information, etc.

**ARThUR**
King Arthur commands his loyal knight Sir Perceval managing the tasks to retrieve data for analysis. It manages data incremental update, parallel downloading, etc.

**KIBITER**
Custom fork of Kibana to work on new ideas for metrics & data visualization to be used by GrimoireLab Panels.

**GRIMOIREELK**
Playground for testing the whole set of tools as a platform for software development analytics. It’s a prototype of the Grimoire Open Development Analytics platform.

**PANELS**
Set of pre-defined widgets and dashboard templates to visualize Elasticsearch indexes generated by GrimoireELK with Kibiter.

**MORDRED**
Docker container to help you deploying the Grimoire Lab using a set of configuration files.

Example of a dashboard:
[.opendaylight.biterg.io](http://opendaylight.biterg.io)
How? Software Committee
GrimoireLab
How? Software Committee
Live demos, proofs of concept

GrimoireLab:
opnfv.biterg.io
cauldron.io

Prospector:
prospector.bitergia.net

Cregit:
cregit.linuxsources.org/
Building a Community

You're welcome to join!!
Building a Community
How to Join

Mailing Lists
Periodic Meetings
Weekly Hangouts
IRC Channels
Issues
Pull requests

cnaoss.community
Let's work together to extract knowledge from CHA OSS!!!
License & Credits

(c) 2017 CHA OSS. Some rights reserved. This presentation is distributed under the “Attribution-ShareAlike 4.0” license, by Creative Commons, available at creativecommons.org/licenses/by-sa/4.0/

- “Napoleon’s Russian campaign of 1812”, Original by Charles Minard
  License: Public domain
  en.wikipedia.org/wiki/Charles_Joseph_Minard#/media/File:Minard.png

- “Aged Come In We’re Open”, Picture by Czarina Alegre in Flickr
  License: Creative Commons Attribution 2.0
  flic.kr/p/fjGamh

- Good code”, Comic by Randall Munroe, XKCD 844
  License: Creative Commons Attribution-NonCommercial 2.5
  xkcd.com/844/