Measuring Success in Soft. Development Projects

Open Leadership Summit, Tahoe 2017

Jesús González Barahona @jgbarah
Daniel Izquierdo Cortázar @dizquierdo
https://speakerdeck.com/bitergia
Outline

Introduction
Open Source Goals
Linux Foundation analytics as use case
Inner Source vs Open Source
Measuring Inner Source
Like five years ago I was having coffees with the gang of Bitergia founders. Involved in the company since then.

My two hats:

I work at Universidad Rey Juan Carlos... ...researching about software development.

Jesus

bitergia.com
gsy.c.es/~jgb
I only have one hat

Bitergia co-founder

OSS researcher

Data analytics

Diversity analysis

Love metrics
Software Development Analytics
for your peace of mind
Introduction
Decisions based on data
Why do we need metrics?

- Check ongoing work
  - Awareness
  - Understanding
- Lead process improvement
  - Migrating to new infrastructure
  - New rules when code reviewing
- Motivational actions
  - Developers following some track - welcome and recognize new contributions
Several dimensions to measure:

- Activity
- Community
- Performance
- Code
- License compliance
Open Source Goals
OSS Goals

“...accelerate open technology development and commercial adoption...”

“...global development, distribution and adoption of the OpenStack cloud...”

“...open, collaborative software development projects...”
Each project has its own mission, but in general:

- Promote adoption and collaboration of their specific products
- Other potential reasons:
  - Become a standard in the industry
  - Free alternative to proprietary soft
  - Philosophical and ethical approach
  - And many other reasons to contribute to free software
It’s all about the **people** using and developing those products

Success = used and developed, by individuals or by the industry

Metrics are used for **transparency**, **neutrality**, **marketing**, and **engineering**
Linux Foundation Analytics
~50 million commits
~80,000 different authors
~7,000 git repositories
~250 mailing lists
~1 million messages
### Git
- **Commits**: 308,538
- **Authors**: 9,238
- **Repositories**: 1,810

### Mailing Lists
- **Emails**: 80,499
- **Participants**: 1,900
- **Mailing Lists**: 174

### Commits
- Time series chart showing number of commits over time.

### Authors
- Time series chart showing number of authors over time.

### Projects
<table>
<thead>
<tr>
<th>Project</th>
<th>Commits</th>
<th>Authors</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>CodeAurora</td>
<td>203,930</td>
<td>4,885</td>
<td>3</td>
</tr>
<tr>
<td>Tizen</td>
<td>20,725</td>
<td>1,869</td>
<td>5</td>
</tr>
<tr>
<td>YoctoProject</td>
<td>18,121</td>
<td>1,344</td>
<td>5</td>
</tr>
<tr>
<td>CloudFoundry</td>
<td>13,495</td>
<td>311</td>
<td>2</td>
</tr>
<tr>
<td>CloudNativeComputingFoundation</td>
<td>9,585</td>
<td>659</td>
<td>5</td>
</tr>
<tr>
<td>XenProject</td>
<td>5,811</td>
<td>347</td>
<td>3</td>
</tr>
<tr>
<td>Linux</td>
<td>5,702</td>
<td>1,057</td>
<td>2</td>
</tr>
<tr>
<td>Dronecode</td>
<td>5,199</td>
<td>139</td>
<td>4</td>
</tr>
<tr>
<td>OpenDaylight</td>
<td>4,956</td>
<td>220</td>
<td>6</td>
</tr>
<tr>
<td>Node.js</td>
<td>4,678</td>
<td>337</td>
<td>3</td>
</tr>
</tbody>
</table>
Git Analytics

The Linux Foundation

Overview  Git  Making Life  Data Store  About

Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Commits</th>
<th>Authors</th>
<th>Repositories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Foundry</td>
<td>2,954,100</td>
<td>18,600</td>
<td>128</td>
</tr>
<tr>
<td>Yastra</td>
<td>932,849</td>
<td>6,370</td>
<td>100</td>
</tr>
<tr>
<td>TOS</td>
<td>3,504,724</td>
<td>10,207</td>
<td>1,893</td>
</tr>
<tr>
<td>XenProject</td>
<td>430,354</td>
<td>5,124</td>
<td>108</td>
</tr>
<tr>
<td>Linux</td>
<td>1,015,355</td>
<td>5,146</td>
<td>3</td>
</tr>
<tr>
<td>Dremocoder</td>
<td>64,603</td>
<td>689</td>
<td>77</td>
</tr>
<tr>
<td>OpenDaylight</td>
<td>38,028</td>
<td>694</td>
<td>75</td>
</tr>
<tr>
<td>Cloud Native Computing Foundation</td>
<td>51,061</td>
<td>2,666</td>
<td>75</td>
</tr>
<tr>
<td>Nestle</td>
<td>58,164</td>
<td>1,382</td>
<td>93</td>
</tr>
</tbody>
</table>

Export: Raw  | Formatted

Bitergia

Metrics

- Commits
- Authors
- Repositories

Time Series

- Commits
- Authors
- Repositories

Graphs

- Node Size
- Edge Size
Mailing Lists

- Bitergia
Project dashboards

Example: OPNFV

http://opnfv.biterg.io
Open Source and Inner Source
IS

SILOS!
IS Goals

Inner source aims at bringing OSS method to the enterprise
Inner source aims at bringing OSS method to the enterprise

Some advantages:

- Reduce time to market
- Share costs and maintenance
- Engagement
- Increase code quality (code review, CI)
- Allow innovation
OSS vs IS

Infrastructure

Dev. Methodology

Open Source

Inner Source
OSS vs IS

- Infrastructure
  - Gitlab, GitHub Enterprise, Atlassian, in house services, mailing lists
- Dev. Methodology
  - Code review, CI, Dev. documentation, governance, meritocracy
- Open Source
- Inner Source
**OSS vs IS**

Inner source is not open source! (but they’re similar)

Some examples

<table>
<thead>
<tr>
<th>Open source</th>
<th>Inner source</th>
</tr>
</thead>
<tbody>
<tr>
<td>● OSS license</td>
<td>● Deal with licenses</td>
</tr>
<tr>
<td>● Open development</td>
<td>● Open development in house</td>
</tr>
<tr>
<td>● Anyone is welcome</td>
<td>● Anyone in the org. Is welcome</td>
</tr>
<tr>
<td>● Foster adoption</td>
<td>● Foster internal use and reusability</td>
</tr>
</tbody>
</table>
Measuring Inner Source
Different initial goals in open and inner source projects.
But, similar development method and infrastructure!
And, similar analysis.
Most of the OSS metrics are useful for IS communities
Let’s measure!
Attraction/Retention

Attracted Devs. → Awesome Project! → Devs. leaving the community
Attraction/Retention

- How good is the community attracting/retaining devs?
  - Number of newcomers
  - Number of retaining devs
- Understanding how some policies affect the attraction/retention rate
Mentorship
Mentorship and helping newcomers

- Mentors are key to help newcomers
- Who are they? And their workload?
- Does the community need more mentors?
- How many people are leading?
Contributors Funnel

From users to core reviewers
Contributors Funnel

- Help to understand how the community evolves
- From the first traces (e.g., email) to become a core reviewer
  - How long does it take?
  - What % of people reach that core level?
Development Cycle

- This helps to measure the time since the user story till the code is merged
  - How fast is the process?
  - Median time to merge, iterations, developers involved, CI, code review bottlenecks
- We know the time to deployment, and the time to close a user story brings the whole picture
Spreading the Knowledge

- Turnover happens
- How are developers connected?
- Fill orphaned areas left by a senior developer
- Territoriality: files touched by just one developer
Some anti-patterns

Do not measure people unless you want to (undesired situations)

- ‘Tell me how you measure me, and I will tell you how I will behave’ - Eliyahu Goldratt, The Haystack Syndrome

Team performance, not people
Conclusions
Summary

Inner source can be compared to OSS projects

You can benchmark your performance with any OSS project of reference (TLF, ASF, OpenStack)

Inner source can learn a lot from OSS (and vice versa)

Success depends on the goals of your organization (but you can benchmark!)

Dashboards are useful to lead that process improvement
Measuring Success in Soft. Development Projects

Open Leadership Summit, Tahoe 2017

Jesús González Barahona @jgbarah
Daniel Izquierdo Cortázaro @dizquierdo
https://speakerdeck.com/bitergia