



# Metrics

Release 2021-03

<https://chaoss.community/metrics>

MIT License

Copyright © 2021 CHAOSS a Linux Foundation® Project

# 内容

<b>CHAOSS Contributors</b>	<b>2</b>
<b>CHAOSS Governing Board Members</b>	<b>3</b>
<b>Common Metrics WG</b>	<b>4</b>
关注领域 - What	5
技术分支	6
贡献类型	8
关注领域 - When	10
活跃的日期和时间	11
突发性波动	14
变更请求中审查周期的持续时间	17
关闭时长	18
第一响应时长	20
关注领域 - Who	22
贡献者位置	23
贡献者	26
组织多元化	30
<b>Value WG</b>	<b>33</b>
关注领域 - 公共价值	34
项目发展速度	35
关注领域 - Placeholder	37
就业机会	38
组织项目技能需求	39
关注领域 - 人力投资	45
人力投资	46
<b>Release History</b>	<b>48</b>
Continuous Metric Contributions Since Last Release	49
Release 2021-03 Notes:	50
Release 2020-08 Notes:	51
Release 2020-01 Notes:	52
Release 2019-08 Notes	53
<b>LICENSE</b>	<b>54</b>

# CHAOSS Contributors

---

Aastha Bist, Abhinav Bajpai, Ahmed Zerouali, Akshara P, Akshita Gupta, Amanda Brindle, Anita Ihuman, Alberto Martín, Alberto Pérez García-Plaza, Alexander Serebrenik, Alexandre Courouble, Alolita Sharma, Alvaro del Castillo, Ahmed Zerouali, Amanda Casari, Amy Marrich, Ana Jimenez Santamaria, Andre Klapper, Andrea Gallo, Andy Grunwald, Andy Leak, Aniruddha Karajgi, Anita Sarma, Ankit Lohani, Ankur Sonawane, Anna Buhman, Armstrong Foundjem, Atharva Sharma, Ben Lloyd Pearson, Benjamin Copeland, Beth Hancock, Bingwen Ma, Boris Baldassari, Bram Adams, Brian Proffitt, Camilo Velazquez Rodriguez, Carol Chen, Carter Landis, Chris Clark, Christian Cmeheil-Warn, Damien Legay, Dani Gellis, Daniel German, Daniel Izquierdo Cortazar, David A. Wheeler, David Moreno, David Pose, Dawn Foster, Derek Howard, Don Marti, Drashti, Duane O'Brien, Dylan Marcy, Eleni Constantinou, Elizabeth Barron, Emily Brown, Emma Irwin, Eriol Fox, Fil Maj, Gabe Heim, Georg J.P. Link, Gil Yehuda, Harish Pillay, Harshal Mittal, Henri Yandell, Henrik Mitsch, Igor Steinmacher, Ildiko Vancsa, Jacob Green, Jaice Singer Du Mars, Jaskirat Singh, Jason Clark, Javier Luis Cánovas Izquierdo, Jeff McAffer, Jeremiah Foster, Jessica Wilkerson, Jesus M. Gonzalez-Barahona, Jilayne Lovejoy, Jocelyn Matthews, Johan Linåker, John Coghlan, John Mertic, Jon Lawrence, Jonathan Lipps, Jono Bacon, Jordi Cabot, Jose Manrique Lopez de la Fuente, Joshua Hickman, Joshua R. Simmons, Josianne Marsan, Justin W. Flory, Kate Stewart, Katie Schueths, Keanu Nichols, Kevin Lumbard, King Gao, Kristof Van Tomme, Lars, Laura Dabbish, Laura Gaetano, Lawrence Hecht, Leslie Hawthorne, Luis Cañas-Díaz, Luis Villa, Lukasz Gryglicki, Mariam Guizani, Mark Matyas, Martin Coulombe, Matthew Broberg, Matt Germonprez, Matt Snell, Michael Downey, Miguel Ángel Fernández, Mike Wu, Neil Chue Hong, Neofytos Kolokotronis, Nick Vidal, Nicole Huesman, Nishchith K Shetty, Nithya Ruff, Nuritzi Sanchez, Parth Sharma, Patrick Masson, Peter Monks, Pranjal Aswani, Pratik Mishra, Prodomos Polychroniadis, Quan Zhou, Ray Paik, Remy DeCausemaker, Ria Gupta, Richard Littauer, Robert Lincoln Truesdale III, Robert Sanchez, Rupa Dachere, Ruth Ikegah, Saicharan Reddy, Saloni Garg, Saleh Abdel Motaal, Samantha Lee, Samantha Venia Logan, Samson Goddy, Santiago Dueñas, Sarit Adhikari, Sarvesh Mehta, Sarah Conway, Sean P. Goggins, Shane Curcuru, Sharan Foga, Shaun McCance, Shreyas, Silona Bonewald, Sophia Vargas, Sri Ramkrishna, Stefano Zacchiroli, Stefka Dimitrova, Stephen Jacobs, Tharun Ravuri, Thom DeCarlo, Tianyi Zhou, Tobie Langel, Saleh Abdel Motaal, Tom Mens, UTpH, Valerio Cosentino, Venu Vardhan Reddy Tekula, Vicky Janicki, Victor Coisne, Vinod Ahuja, Vipul Gupta, Will Norris, Xavier Bol, Xiaoya, Zibby Keaton

Are you eligible to be on this list? You are if you helped in any capacity, for example: Filed an issue. Created a Pull Request. Gave feedback on our work. Please open an issue or post on the mailing list if we've missed anyone.

## CHAOSS Governing Board Members

---

- Amy Marrich, Red Hat
- Andrea Gallo, Linaro
- Armstrong Foundjem, MCIS Laboratory at Queen's University
- Daniel Izquierdo, Bitergia
- Daniel M. German, University of Victoria
- Dawn Foster, VMware
- Don Marti, CafeMedia
- Georg Link, Bitergia
- Ildikó Vancsa, OpenStack
- Kate Stewart, Linux Foundation
- Matt Germonprez, University of Nebraska at Omaha
- Nicole Huesman, Intel
- Ray Paik, GitLab
- Sean Goggins, University of Missouri
- Wayne Beaton, Eclipse Foundation

# Common Metrics WG

关注领域	目标
What	了解组织和个人正在做出哪些贡献。
When	了解来自组织 and 个人的贡献何时发生。
Who	了解组织和个人参与开源项目。

## 关注领域 - What

目标: 了解组织和个人正在做出哪些贡献。

度量指标	问题
技术分支 贡献类型	在代码开发平台一个开源项目有多少技术分支? 正在进行哪些类型的贡献?

## 技术分支

问题: 在代码开发平台一个开源项目有多少技术分支?

### 描述

一个技术分支是一个项目分布式版本控制的拷贝。技术分支数表示在同一代码开发平台上该项目的拷贝数。

### 目标

技术分支度量的目标是确定在代码开发平台上存在的该项目的拷贝数。对技术分支的分析可以提供对分支意图的深入了解 (建立分支的不同意图, 如对外游有贡献的分支和没有贡献的分支)。

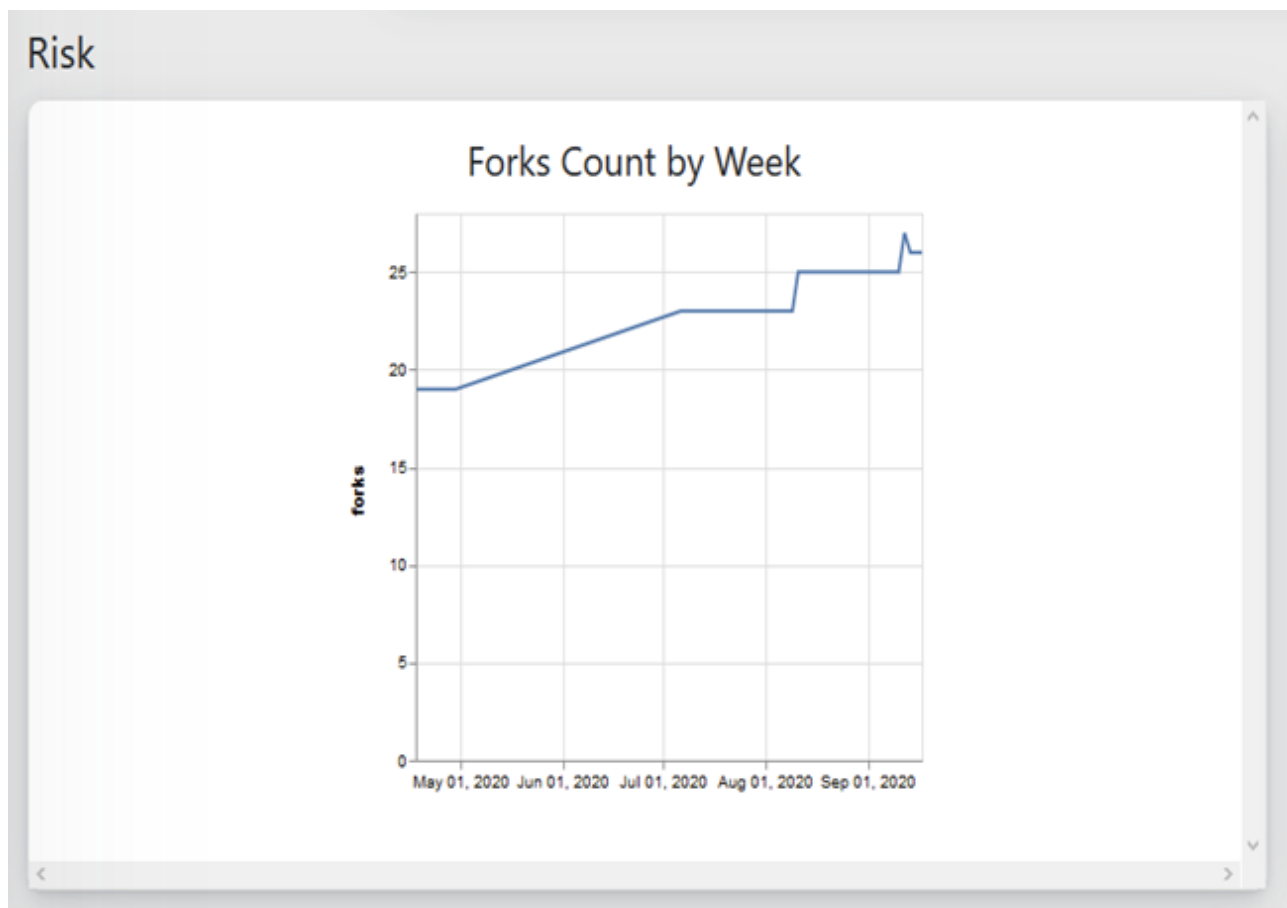
### 实现

#### 筛选条件

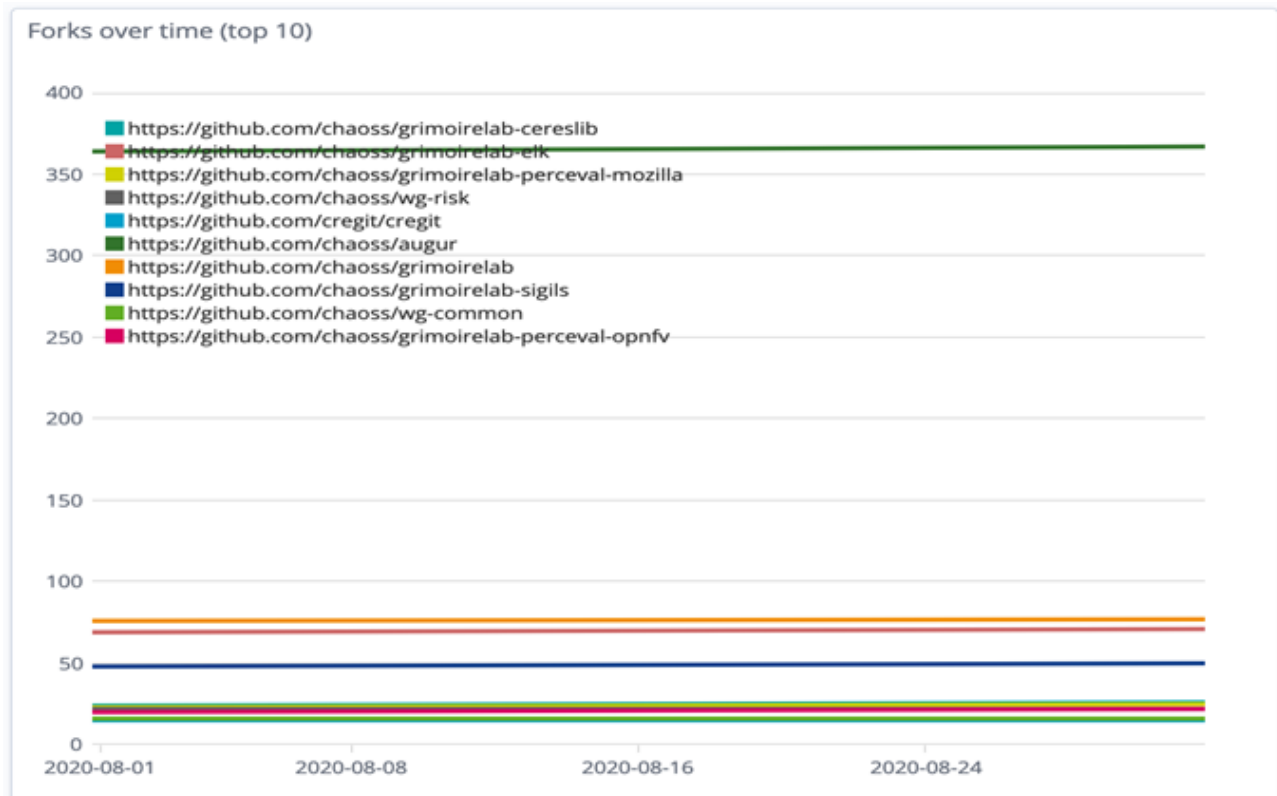
- 时间周期 (例如, 每周, 每月, 每年)
- 贡献分支与总分支的比率 (贡献分支是指对上游代码仓库建立更改请求的分支。)
- 非贡献分支与总分支的比率 (贡献分支是指从未对上游代码仓库建立更改请求的分支。)

### 可视化效果

#### Augur 实现



## GrimoireLab 实现



## 提供度量的工具

- Augur
- GrimoireLab

## 数据收集策略

### Github API

<https://developer.github.com/v3/repos/forks/#list-forks>

### GitLab API

<https://docs.gitlab.com/ee/api/projects.html#list-forks-of-a-project>

### Bitbucket API

[https://developer.atlassian.com/bitbucket/api/2/reference/resource/repositories/%7Bworkspace%7D/%7Brepo\\_slug%7D/forks](https://developer.atlassian.com/bitbucket/api/2/reference/resource/repositories/%7Bworkspace%7D/%7Brepo_slug%7D/forks)

## 参考资料

<https://help.github.com/en/enterprise/2.13/user/articles/fork-a-repo> <https://opensource.com/article/17/12/fork-clone-difference>



## 贡献类型

问题：正在进行哪些类型的贡献？

### 描述

多元化的贡献能够使开源项目健康发展。在很多项目中，有些社区成员并不编写代码，但他们同样做出了有价值的贡献，比如管理社区、区分错误、宣传项目、支持用户或以其他方式提供帮助。

### 目标

多样的贡献类型表明项目是成熟全面的，包含足够的活动来支持项目的所有方面，并且提供多种贡献类型的晋升渠道，让拥有编码之外的不同专长的人员也能够发展到领导层。

### 实现

如何对贡献进行定义、量化、跟踪和公布是一个具有挑战性的问题。每个项目的答案可能都是独一无二的，以下建议仅作为抛砖引玉。作为一个通用指南，很难将不同的贡献类型相互比较，应单独考量。

- 以下列表可以帮助确定贡献类型：
  - 编写代码
  - 审查代码
  - 错误分类
  - 质量保证和测试
  - 安全相关活动
  - 本地化和翻译
  - 事件组织
  - 文档编写
  - 社区建设和管理
  - 教学和教程构建
  - 故障排除和支持
  - 创意作品和设计
  - 用户界面、用户体验和易用性
  - 社交媒体管理
  - 用户支持和问题解答
  - 撰写文章
  - 公共关系 - 技术媒体采访
  - 事件发言
  - 营销与活动宣传
  - 网站开发

- 法律委员会
- 财务管理

## 数据收集策略

- \*\* 采访或调查： \*\* 让社区成员认可他人的贡献，找出过去没有考虑到的贡献类型。
  - 您想表彰谁在项目中的贡献？他们贡献了什么？
- \*\* 观察项目： \*\* 找出和认可项目不同部分的负责人。
  - 在项目网站或代码仓库中列出了哪些负责人？
- \*\* 捕获非代码贡献： \*\* 通过问题跟踪器等专用系统跟踪贡献。
  - 日志记录可以包含项目要了解的关于非代码贡献的定制化信息，用以识别工作量。
  - 通过沟通渠道活动的代理贡献。例如，如果质量保证成员 (QA) 拥有自己的邮件列表，就可以通过邮件列表活动来代理衡量围绕 QA 贡献的活动。
- \*\* 收集跟踪数据： \*\* 通过协作工具日志数据衡量贡献。
  - 例如，可以从源代码仓库计算代码贡献，可以从维基编辑历史记录计算维基贡献，可以从电子邮件归档计算电子邮件消息
- \*\* 自动分类： \*\* 训练人工智能 (AI) 机器人来检测贡献并对其分类。
  - AI 机器人可以协助对贡献进行分类，例如，帮助请求与提供的支持，或功能请求与错误报告，尤其是以上均在同一个问题跟踪器中完成的情况。

其他考量：

- 特别是对于自动报告，允许社区成员选择退出并不出现在贡献报告上。
- 承认对贡献类型的捕捉不完善，并明确说明收集了哪些类型的贡献。
- 随着项目发展，贡献类型的收集方法需要作出调整。例如，交换国际化库时，围绕本地化的项目活动可能会在变化前后产生不同的指标。
- 大规模挖掘贡献类型时，要考虑机器人的活动。

## 参考资料

- <https://medium.com/@sunnydeveloper/revisiting-the-word-recognition-in-foss-and-the-dream-of-open-credentials-d15385d49447>
- <https://24pullrequests.com/contributing>
- <https://smartbear.com/blog/test-and-monitor/14-ways-to-contribute-to-open-source-without-being/>
- <https://wiki.openstack.org/wiki/AUCRecognition>
- <https://www.drupal.org/drupalorg/blog/a-guide-to-issue-credits-and-the-drupal.org-marketplace>

## 关注领域 - When

**目标:** 了解来自组织和个人的贡献何时发生。

度量指标	问题
活跃的日期和时间	贡献者活跃的日期和时间戳是什么时候？
突发性波动	如何在项目中观察到一个活动在短时间的突发性波动，然后相应地恢复到这个活动的正常状态？
变更请求中审查周期的持续时间	单个变更请求中审查周期的持续时间是多少？
关闭时长	创建和关闭操作（如议题、更改请求或需要支持的问题单）之间需要多少时间？
第一响应时长	对于需要关注的活动，从创建到第一次响应需要多长时间？

## 活跃的日期和时间

问题：贡献者活跃的日期和时间戳是什么时候？

### 描述

个人在一天在不同时间参与开源项目活动。此指标旨在确定个人活动完成的日期和时间。在非 UTC 时区，数据可用于概率性估计贡献来自地球哪个地方。

### 目标

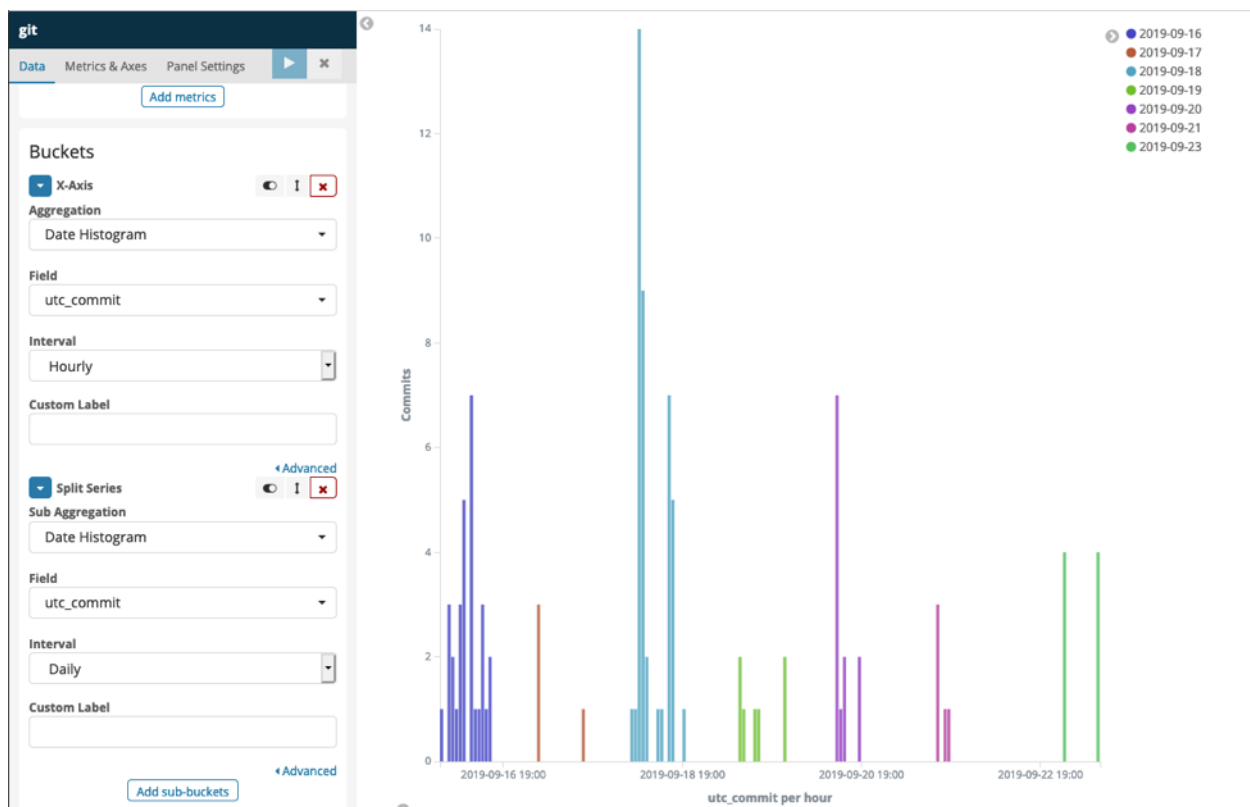
- 为雇主提高透明度，帮助其了解组织员工参与开源项目的时间
- 为开源项目 and 社区管理者提高透明度，帮助其了解活动发生的时间

### 实现

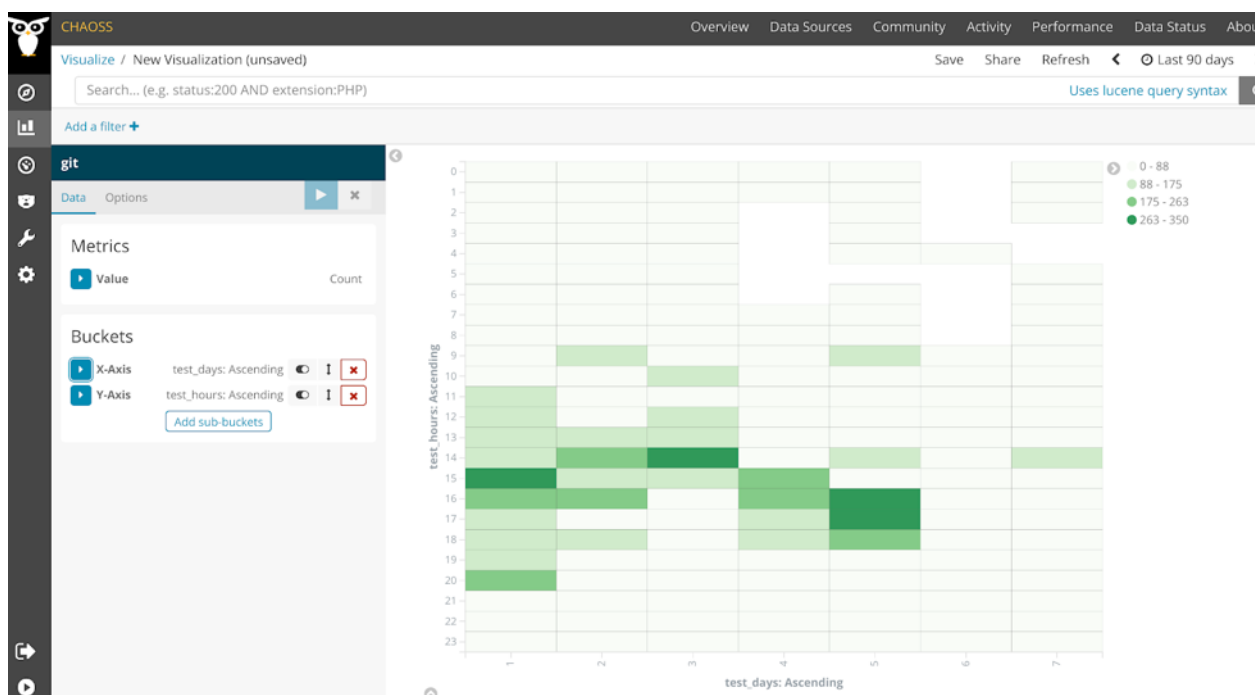
#### 筛选条件

- 通过组织筛选个人
- 按 UTC 时间聚合
  - 可以显示全球贡献时间分布；项目最活跃时间。
- 按当地时间聚合
  - 可以显示贡献者做贡献时的当地时间。可以得到结论贡献者是在工作时间贡献更多，还是在晚间（非工作时间）贡献更多。
- 代码仓库 ID
- 社区细分（例如，GrimoireLab 在欧盟时区更活跃，Augur 在美国时区更活跃）

#### 可视化效果



Time	author_name	repo_name	tz	utc_author	utc_commit	commit_date
Sep 23rd 2019	Carter Landis	https://github.com/chaoss/augur	-5	September 23rd 2019, 15:42:16.000	September 23rd 2019, 15:42:16.000	September 23rd 2019, 10:42:16.000
Sep 23rd 2019	Matt Snell	https://github.com/chaoss/augur	-5	September 23rd 2019, 13:21:56.000	September 23rd 2019, 13:21:56.000	September 23rd 2019, 08:21:56.000
Sep 23rd 2019	Gabe Heim	https://github.com/chaoss/augur	-5	September 23rd 2019, 12:31:49.000	September 23rd 2019, 12:31:49.000	September 23rd 2019, 07:31:49.000
Sep 23rd 2019	Santiago Duehas	https://github.com/chaoss/grimoirelab-perceval	2	September 23rd 2019, 12:30:18.000	September 23rd 2019, 12:30:18.000	September 23rd 2019, 14:30:18.000
Sep 23rd 2019	Valerio Cosentino	https://github.com/chaoss/grimoirelab-perceval	2	September 23rd 2019, 11:26:04.000	September 23rd 2019, 11:26:30.000	September 23rd 2019, 13:26:30.000
Sep 23rd 2019	Valerio Cosentino	https://github.com/chaoss/grimoirelab-perceval	2	September 23rd 2019, 11:24:21.000	September 23rd 2019, 11:24:21.000	September 23rd 2019, 13:24:21.000





## 提供指标的工具

[GrimoireLab](#)

[Augur](#) 日期/时间戳

## 参考资料

[协调世界时](#)

## 突发性波动

问题: 如何在项目中观察到一个活动在短时间的突发性波动, 然后相应地恢复到这个活动的正常状态?

### 描述

有多种原因可能会促使代码仓库中的某个活动的总量突然增加或减少。这些增加和减少都表现为该活动相对于平均活动量的突然变化。突发性是一种了解现有指标中活动周期的方法, 例如问题、合并请求、邮件列表、代码提交或评论。某个活动突发性变化的根本原因的示例包括:

- 发布周期
- 全球流行病
- 黑客松活动
- 导师计划
- 展示工具的会议、聚会和其他活动
- 传统和社交媒体的公告和提及
- 引起人们高度关注的关键 Bug
- 为解决特定问题的社区设计会议或头脑风暴会议
- 社区成员来自于另一个依赖你项目 (例如, 依赖项) 的社区

### 目标

- 确定活动突发性波动的根本原因的影响
- 当项目活动在不知不觉中上升时提高意识
- 帮助捕捉某个项目活动增加或减少的意义
- 帮助社区和维护者为遵循某种规律的未来突发性波动做好准备
- 帮助衡量有影响力的外部活动对本社区的影响
- 区分非正常活动与正常活动

### 实现

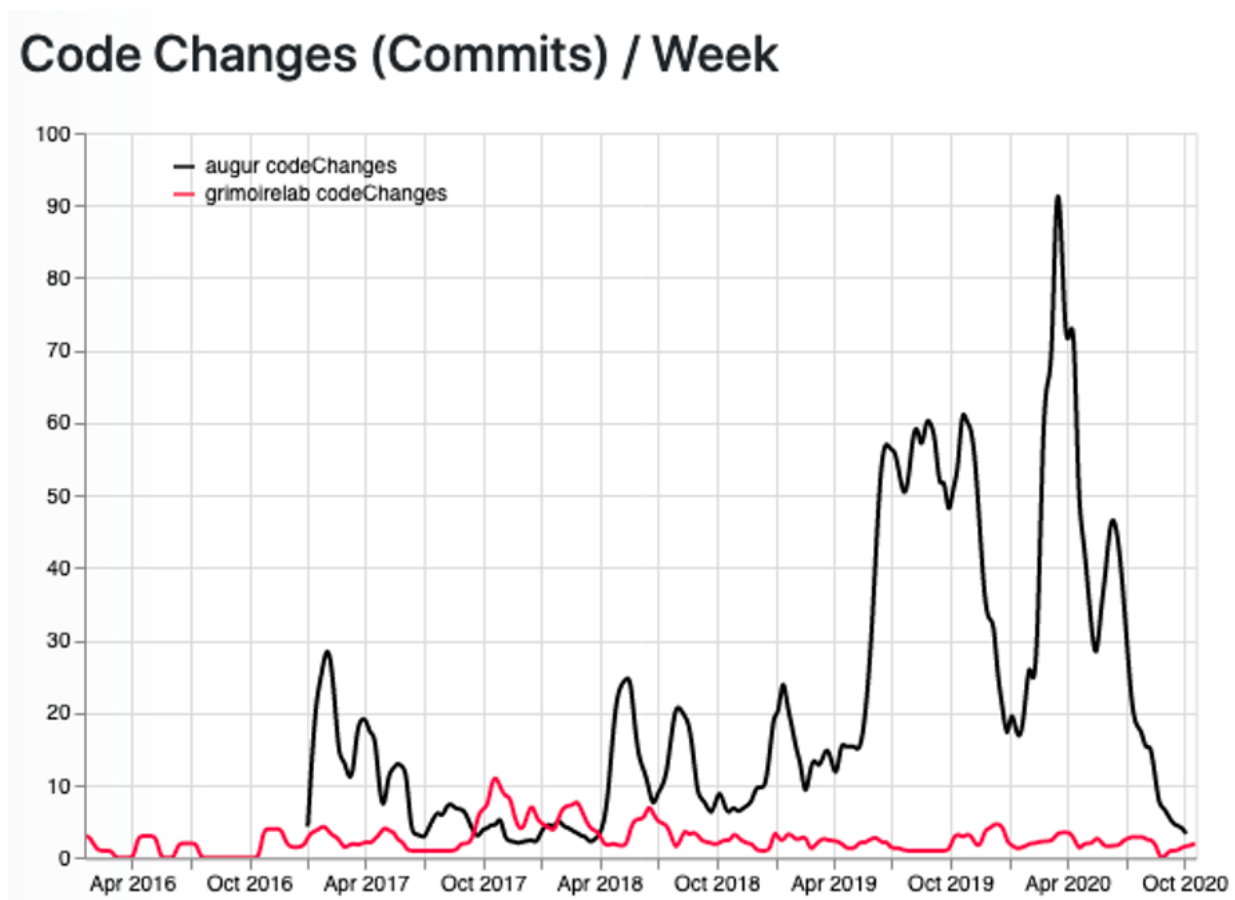
#### 筛选条件

- 标星
- 分支
- 问题或者 bug 报告
- 标签
- 下载
- 发布标签
- 变更请求

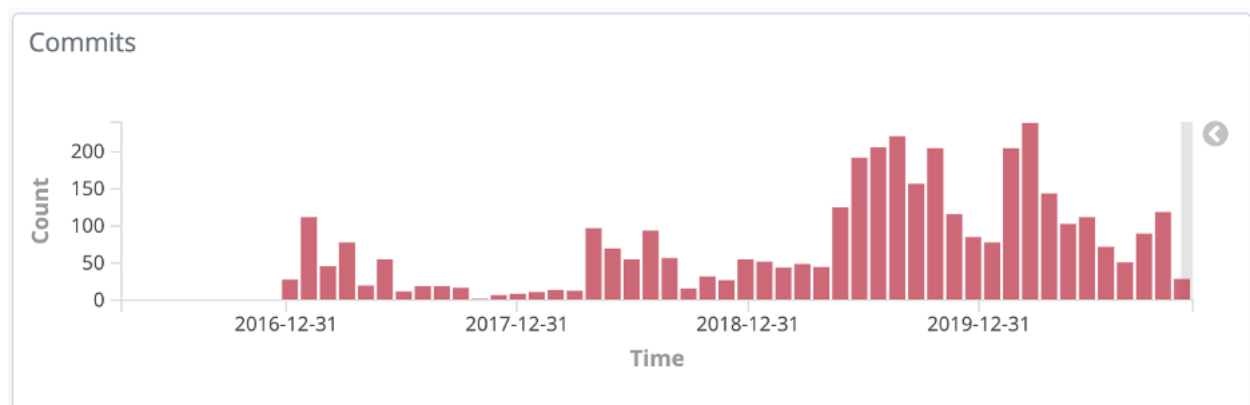
- 邮件列表流量
- 文档添加或修订
- 新建代码仓库
- 新功能请求
- 消息对话
- 传统和社交媒体活动
- 会议出席和提交

## 可视化效果

Augur:



GrimoireLab:





## 提供度量的工具

- Grimoire Lab
- Augur

## 数据收集策略

- 定量
  - 基于时间框识别偏离某种常规状态的活动
  - 某些阈值的异常值，使用布林带等统计数据来衡量稳定性或波动性: [https://en.wikipedia.org/wiki/Bollinger\\_Bands](https://en.wikipedia.org/wiki/Bollinger_Bands)
- 定性调查问题
  - 为什么你在某一段时间内贡献量更多?
  - 你认为这些特有突发性波动的根本原因是什么?
  - 不同事件 (例如, 黑客松, 导师计划, 或者会议) 对项目活动有什么影响?

## References

该指标的灵感来自 Goh 和 Barabasi (2008): <https://arxiv.org/pdf/physics/0610233.pdf>

## 变更请求中审查周期的持续时间

问题: 单个变更请求中审查周期的持续时间是多少?

### 描述

一个变更请求会基于一个或者多个审查周期。在一个审查周期内, 一个或多个审查者可以对提议的贡献提供反馈。审查周期的持续时间, 或贡献的每次新迭代之间的时间, 是该指标的基础。

### 目标

该项目为项目维护者提供的相关洞察:

- 代码审查过程迟滞缓慢, 因为有更多的迭代以及审查周期增加。
- 流程瓶颈导致代码审查迭代时间长。
- 审查周期中维护者或提交者响应缓慢, 导致审查周期处于放弃或者半放弃状态。
- 具有不同审查循环周期长度的评论的特征。

### 实现

审查周期持续时间用于衡量单个变更请求中一个审核周期的时间长度。持续时间可以计算的间隔包括:

- 每个审查周期开始的时刻, 定义为提交或更新变更请求的时间点。
- 每个审查周期结束的时刻, 可能是因为变更请求已更新并需要重新审查, 或者是因为变更请求被接受或拒绝。

### 筛选条件

平均或中位值持续时间, 可以选择性的使用过滤或者分组条件包括:

- 参与审核的人数
- 审核中的评论数
- 变更请求中涉及的修改文件或者代码
- 提出变更请求的项目或组织
- 提交变更请求的时间
- 为变更请求做出贡献的开发人员
- 变更请求
- 变更请求的审核周期数 (例如, 按第一、第二、.....轮过滤)

### 可视化效果

### 提供度量的工具

### 参考资料

可用于实现度量的数据示例: <https://gerrit.wikimedia.org/r/c/mediawiki/core/+194071>

## 关闭时长

问题：创建和关闭操作（如议题、更改请求或需要支持的问题单）之间需要多少时间？

### 描述

关闭时长是指从创建到关闭操作（如议题、更改请求或需要支持的问题单）的总时长。操作需要具有打开和关闭的状态，比如代码审查进程、问答论坛、问题跟踪系统中经常出现的情况。

相关指标：[问题解决时长](#)

### 目标

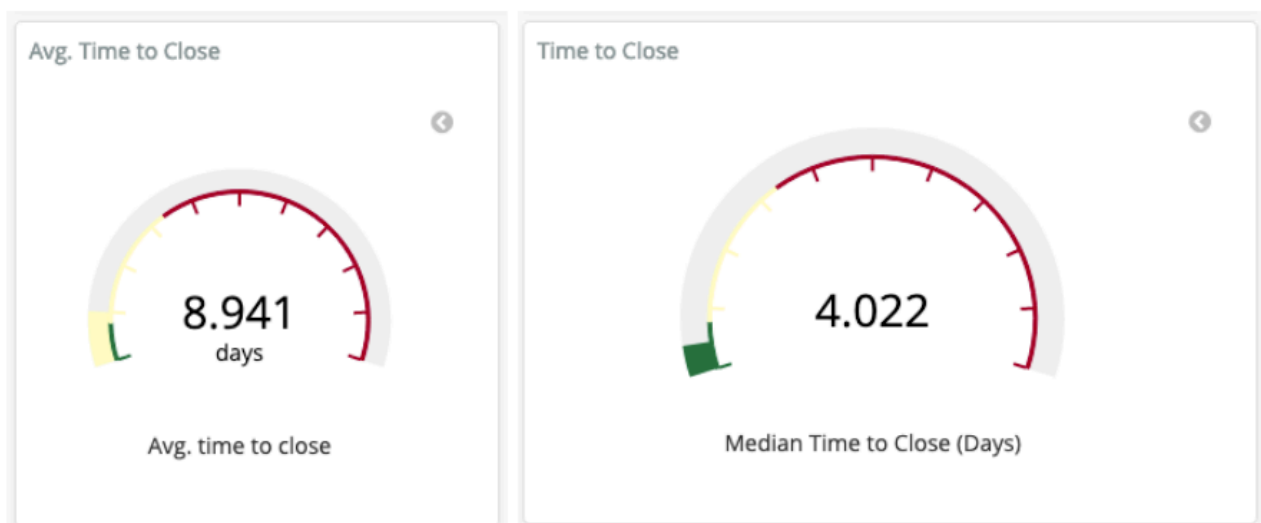
1. 确定社区的响应程度，帮助增加包容性，吸引新贡献者并保留现有贡献者。
2. 找出导致快速或缓慢关闭的操作特征（如寻找最佳实践、改进领域、评估效率）。
3. 识别对不同社区成员及时响应的偏差。
4. 检测社区活动的变化（例如，显示潜在的维护者倦怠、贡献多元化的减少）
5. 了解关闭议题或更改请求的时间与合并成功或失败的关系

### 实现

#### 筛选条件

- 操作的创建者（例如，新贡献者相对于维护者）
- 最初关闭，最后关闭
- 标签（例如错误与新功能）
- 更改请求合并状态（例如，合并时间或没有合并的关闭时间）

### 可视化效果



## 提供指标的工具

Augur 实现：

- 问题解决时长
- 问题持续时间
- 问题响应时间

GrimoireLab 实现：

- 拉取请求效率
- 问题效率
- [Efficiency:TimingOverview](#)

## 数据收集策略

关闭时长指标可根据项目活动和目标的具体情况而定。例如，错误报告的关闭时长可能提供与新功能请求的关闭时长不同的信息。数据收集策略应针对不同的项目目标。可能影响这些进程的其他变量是：

- 问题跟踪系统：如错误报告、蓝图 (OpenStack 专有命名)、用户故事 (user story)、功能请求、epic 等可能会影响事件关闭时长的议题类型。优先级或严重性等其他变量可能有助于推进这一事件的关闭速度。
- 变更请求流程：这取决于变更请求的平台架构，如 Gerrit、GitHub 或邮件列表（如 Linux 内核中），并可能根据进程的复杂程度而有所不同。例如，新人和经验丰富的高级开发者将以不同的方式开展进程，所需时间或多或少。
- 问答论坛：这取决于回答的质量和提问者的意见。有效答案会被标记，提问者成功找到自己问题的正确答案后，进程随即关闭。

## 参考资料

- “Practice P.12: Respond to all submissions”，出自 “Appendix to: Managing Episodic Volunteers in Free/Libre/Open Source Software Communities”，Ann Barcomb、Klaas-Jan Stol、Brian Fitzgerald 和 Dirk Riehle：<https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/13519>

## 第一响应时长

问题：对于需要关注的活动，从创建到第一次响应需要多长时间？

### 描述

对活动的首次响应有时可能是最重要的响应。首次响应表明社区活跃并且积极参与沟通。如果对活动的响应时间较长，可能表示社区不够积极。短时间内对活动做出响应，有助于吸引更多成员参与社区内的进一步讨论。

### 目标

确定各种活动的首次响应节奏，涉及 PR、问题、电子邮件、IRC 帖子等。对于项目的新贡献者、长期贡献者以及整体项目健康情况，首次响应时长都是一项重要考虑因素。

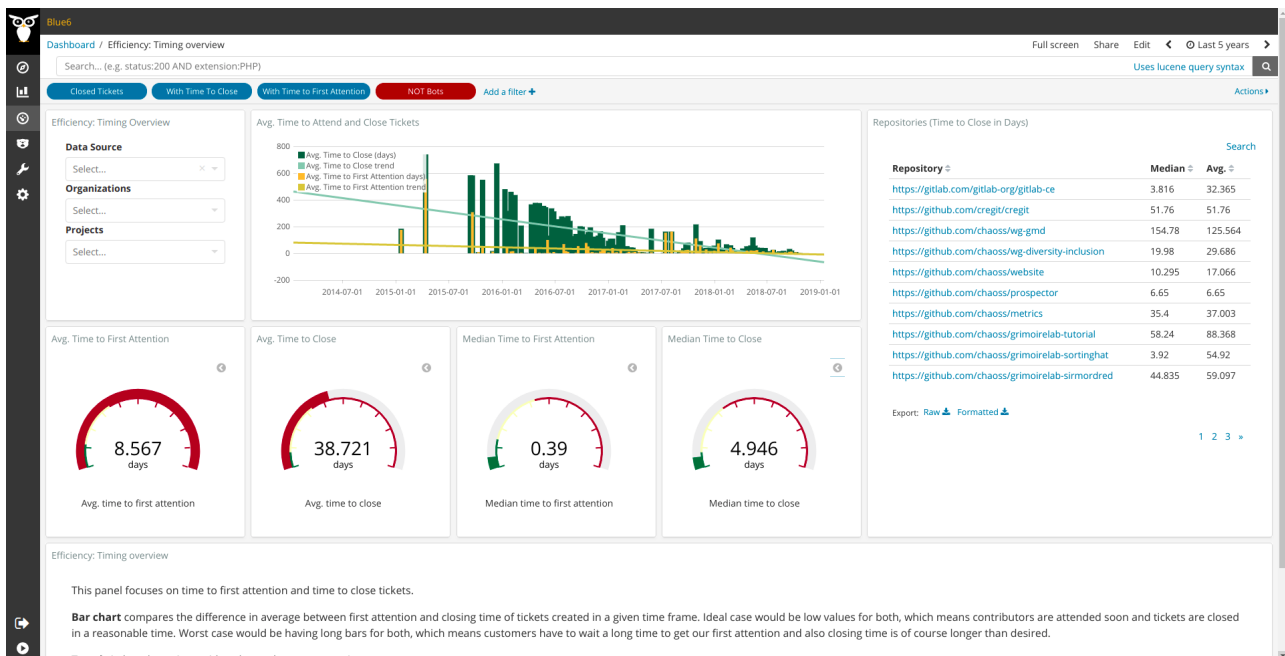
### 实现

活动的首次响应时长 = 首次响应发布到活动的时间 - 活动创建时间。

### 筛选条件

- 回复者的角色，例如，只计算维护者的响应
- 自动响应，例如，通过筛选排除机器人和其他自动回复，只统计真人回复
- 活动类型，如问题（见指标[问题响应时间](#)）、电子邮件、聊天信息、变更请求

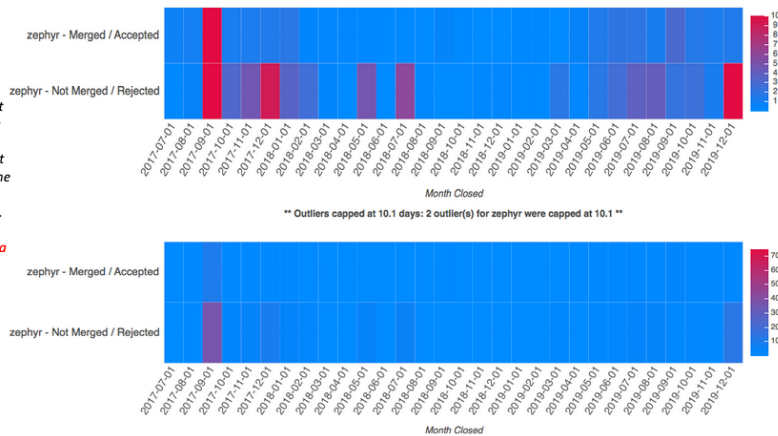
### 可视化效果



## Mean Days to First Response for Closed Pull Requests

Some Internal Slowing, But Outperforming Other Repositories

The heat maps illustrate variance in mean pull request duration to the first response for Zephyr over time. The left side compares Zephyr against itself, and the heat map on the right side is on a scale more aligned with the competition. Note especially that the internal Zephyr gradient has a max of 10 days, while the external comparisons have a max of 75 days.

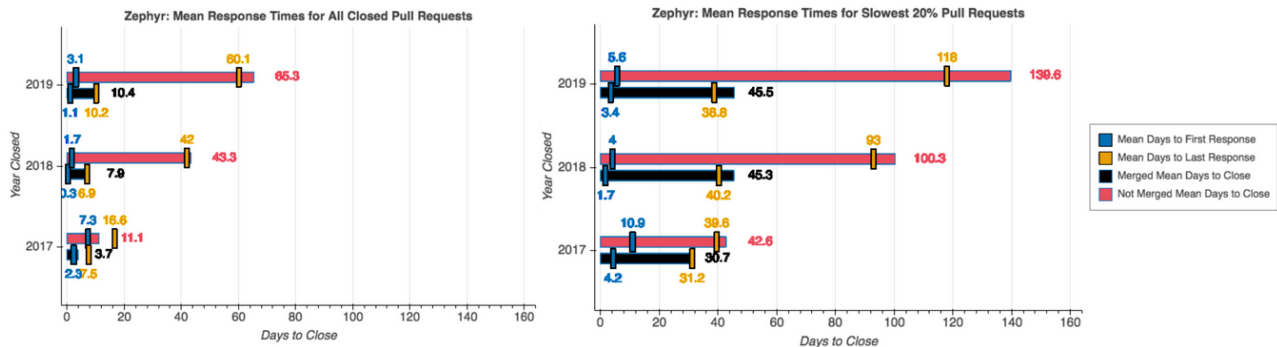


Internal Zephyr PR Performance (0 - 10 scale)

Zephyr PR Performance Compared to Others (0 - 75 scale)

## Mean Response Times (Days) For Closed Pull Requests

Long Running Pull Requests Are Usually Rejected



The length of the black bar illustrates the total number of days that rejected and merged pull requests were open. The blue bar shows that, for the Zephyr project, we see the mean time to first response improving significantly for both rejected and merged pull requests from 2017 - 2019. The orange bar shows the last response or event associated with a pull request.

## 提供指标的工具

- GrimoireLab 面板：效率时序总览
- Kata Containers 仪表板效率面板

## 参考资料

## 关注领域 - Who

目标: 了解组织和个人参与开源项目。

度量指标	问题
贡献者位置	贡献者的位置在哪里?
贡献者	谁是项目的贡献者?
组织多元化	什么是贡献的组织多元化?

## 贡献者位置

问题：贡献者的位置在哪里？

### 描述

贡献者所处的地理位置、居住地或工作地。

### 目标

用于确定贡献者的全球位置，了解他们的工作实践和时区。确定没有贡献的区域，以提高这些区域的参与度；

### 实现

### 筛选条件

按以下条件筛选贡献：

- **\*\* 位置。\*\*** 将各区域的位置分组，以进行多级报告。在此语境中，“位置”是一个故意模糊的术语，可以指地区、国家、州、地域或时区。
- **\*\* 时间段。\*\*** 开始日期和完成日期。默认：永久。计算贡献的时期段。
- **贡献者类型**，例如：
  - 仓库作者
  - 议题作者
  - 代码审查 (Code Review) 参与者
  - 邮件列表作者
  - 事件参与者
  - IRC 作者
  - 博客作者
  - 按发布周期
  - 项目的编程语言
  - 项目中的角色或职能

### 可视化效果

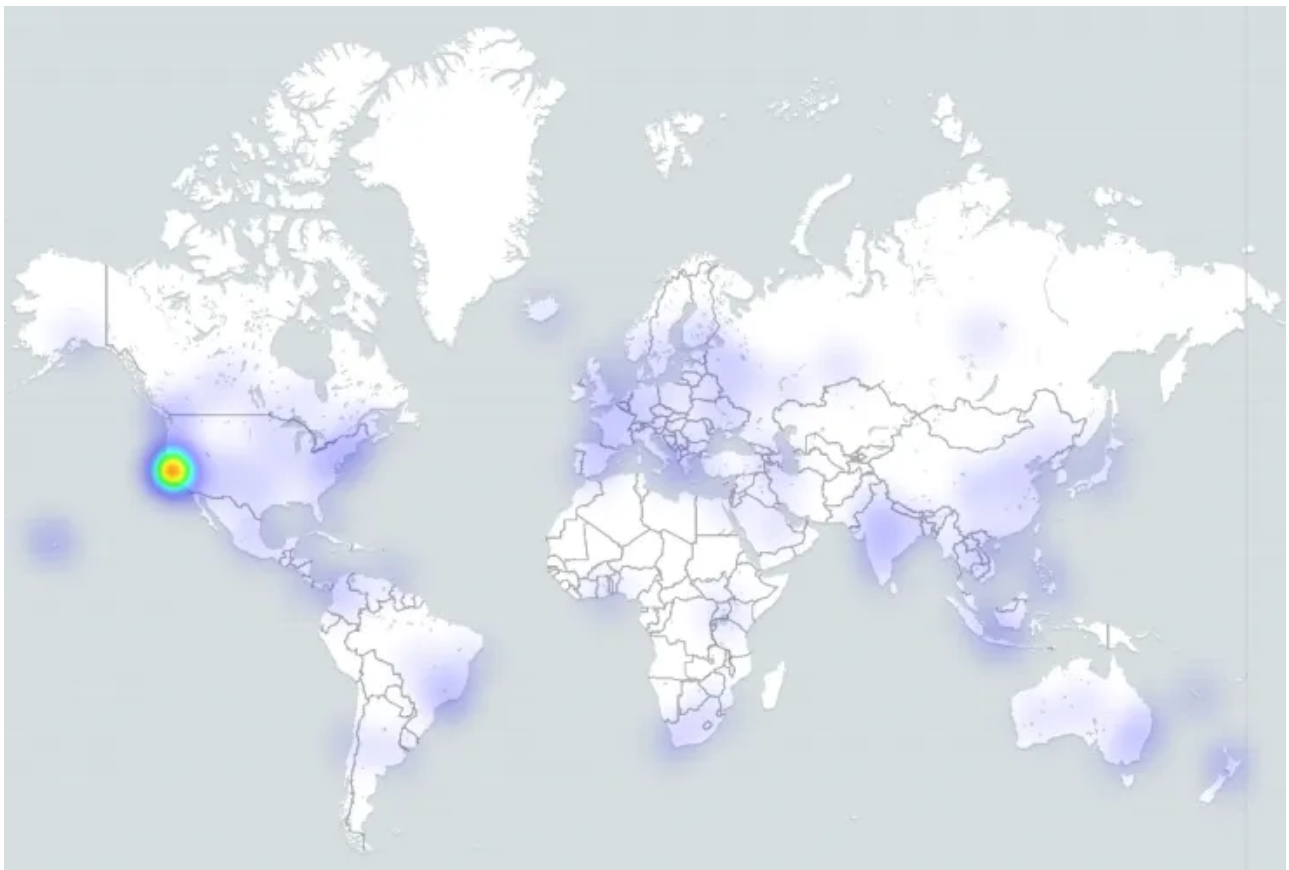
点密度图：





来源: <https://chaoss.biterg.io/goto/a62f3584a41c1c4c1af5d04b9809a860>

可视热图:



来源: <https://blog.bitergia.com/2018/11/20/ubers-community-software-development-analytics-for-open-source-offices>

## 提供度量的工具

- GrimoireLab
- Augur

## 数据收集策略

可以采用不同的方法收集位置信息：

- 在参与度系统中收集贡献者资料的位置信息。
- 使用做出贡献的最频繁位置的 IP 地址地理定位。
- 根据贡献中的时间戳推断地理位置。
- 调查贡献者。

确定贡献者的位置是数据收集的关键挑战。最佳实践是利用参与度系统提供的各种资料信息，如果没有这些信息，可以使用 IP 地理定位来确定该个人最频繁的贡献位置。注意，贡献者可能会在个人资料信息中输入虚假或无意义的位置信息（如“地球”或“互联网”）。注意，IP 地理定位可能会由于 VPN 或其他 IP 屏蔽工具提供大量误报。

另一个考虑因素是外部数据收集工具，如社区调查或事件登记数据，可能交叉引用参与概况的系统。贡献者位置数据可与事件[参与者统计信息](#)和[演讲者统计信息](#)内联收集。

## 参考资料

- Gonzalez-Barahona, J. M., Robles, G., Andradas-Izquierdo, R., & Ghosh, R. A. (2008). Geographic origin of libre software developers. *Information Economics and Policy*, 20(4), 356-363.

## 贡献者

问题：谁是项目的贡献者？

### 描述

贡献者是指以任意方式为项目做出贡献的人。这一指标确保所有类型的贡献在项目中都能得到充分认可。

### 目标

开源项目由许多贡献者组成。认识项目的所有贡献者对于了解个人参与的活动非常重要，比如谁在帮助代码开发、事件规划和营销工作等。

### 实现

从项目使用的协作工具中收集作者姓名。

**聚合器：**

- 计数。给定时间内的贡献者总数。

**参数：**

- 时间段。开始日期和完成日期。默认：永久。计算贡献的时期段。

### 筛选条件

按参与地点。例如：

- 提交作者
- 议题作者
- 审查参与者，例如拉取请求中
- 邮件列表作者
- 事件参与者
- IRC 作者
- 博客作者
- 按发布周期
- 项目活动的时间框架，例如，寻找新贡献者
- 项目的编程语言
- 项目中的角色或职能

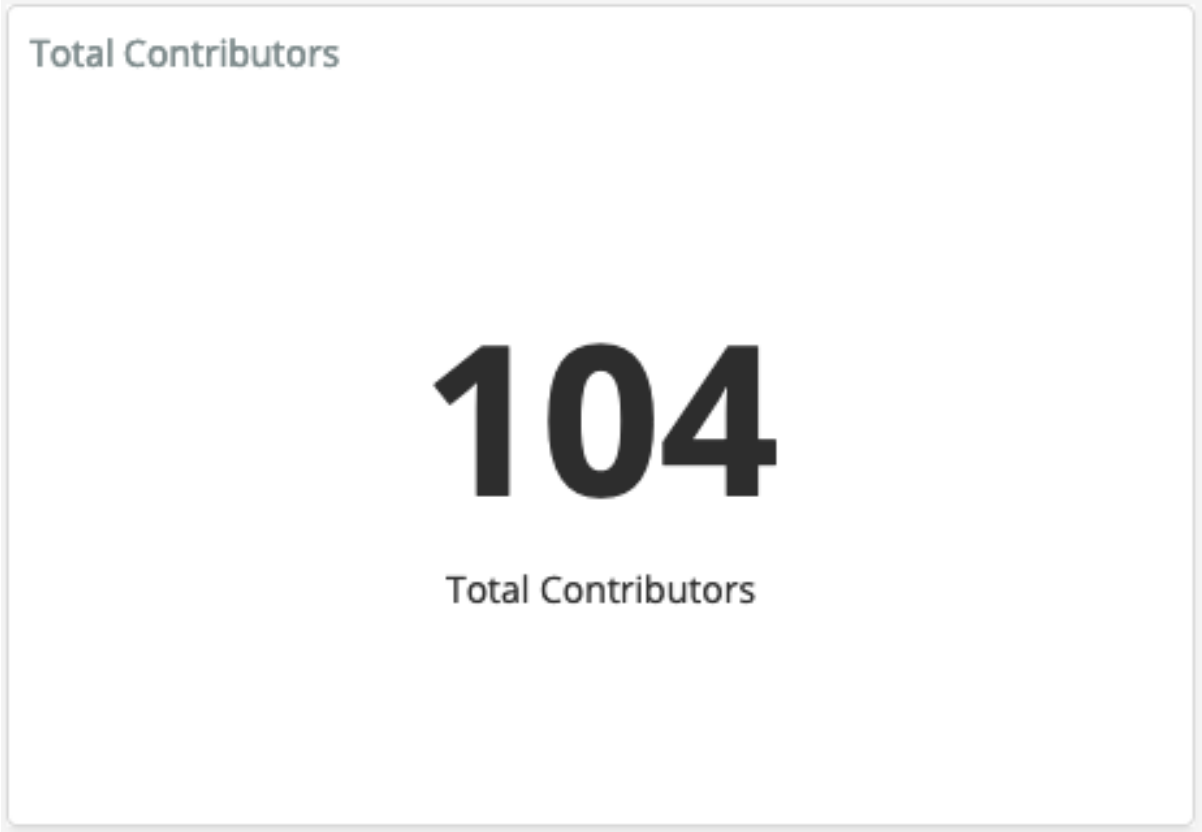
### 可视化效果

1. 贡献者名称列表（通常带有参与程度的信息）

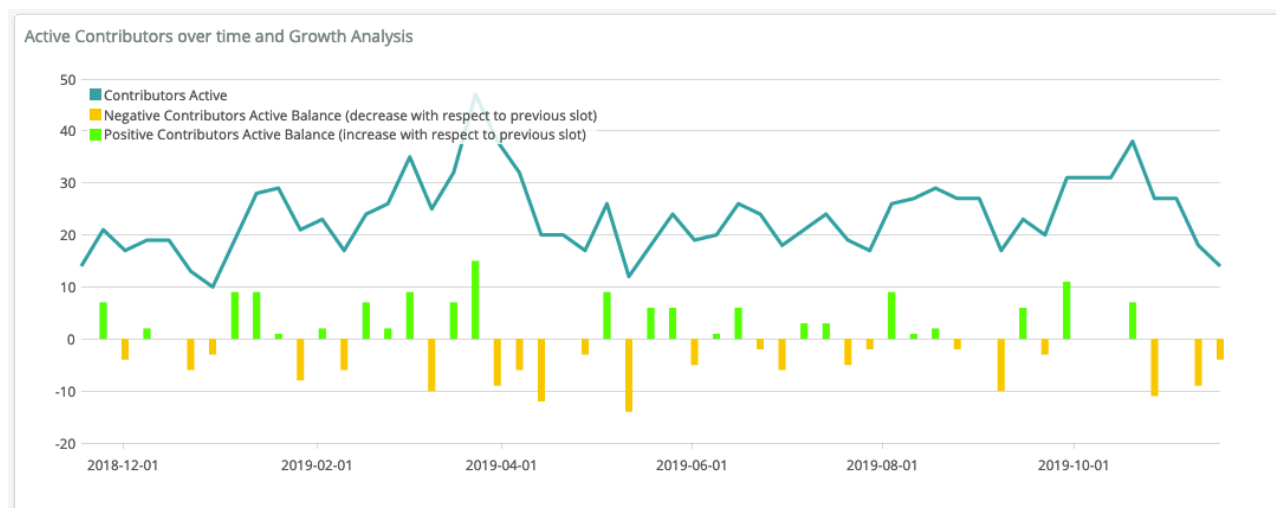
Lines of code added by the top 10 authors

Author	2012	2013	2014	2015	2016	2017	2018
John Doe	0	133	0	3444	37	12905	1361
Jane Smith	0	0	0	0	0	0	59
Bob Johnson	0	0	0	33	0	0	0
Alice Williams	0	0	0	0	0	0	33
Charlie Brown	0	0	0	0	0	17	0
Diana Prince	0	0	0	7	0	0	0
Frank Miller	0	0	0	1	0	0	0

2. 贡献者人数汇总



### 3. 活跃贡献者数量随时间的变化



#### 1. 新贡献者（按首次贡献日期对贡献者排序）

Last Attracted Developers	
Author	First Commit Date
	Apr 9th 2019, 08:47
	Apr 30th 2019, 13:53
	May 5th 2019, 09:35
	May 8th 2019, 08:54
	May 10th 2019, 14:47

## 提供指标的工具

- [GrimoireLab](#)
- [Augur](#)

## 数据收集策略

如上所述，部分贡献者信息可以通过 GrimoireLab 和 Augur 等软件获得。然而，有些贡献者洞察不太容易通过跟踪数据获得。在这些情况下，使用社区成员调查或事件注册的方式可以提供所需信息。示例问题包括：

- 采访问题：通常哪些贡献者不会出现在贡献者列表中？
- 采访问题：哪些贡献者经常不被视为重要贡献者，因为其贡献更倾向于“幕后”？

- 采访问题：您经常与哪些社区成员合作？

此外，社区成员调查可以提供有关项目贡献的更多信息。示例问题包括：

- 李克特量表 [1-x] 项：我正在为项目做贡献
- 矩阵调查项：您在项目中多久参与一次以下活动？
  - 列标题：从不，很少（每月少于一次），有时（每月一次以上），经常（每周一次或多次）
  - 行包括：a) 贡献/审查代码，b) 创建或维护文档，c) 翻译文档，d) 参与项目开发决策，e) 担任社区组织者，f) 指导其他贡献者，g) 亲自参与事件，h) 通过学校或大学计算机项目参与，i) 通过 Outreachy、Google Summer of Code 等计划参与，j) 帮助 ASF 运作（如委员会会议或筹款）

## 参考资料

## 组织多元化

问题：什么是贡献的组织多元化？

### 描述

组织多元化表示一个项目有多少组织参与，以及组织之间不同的参与程度。

### 目标

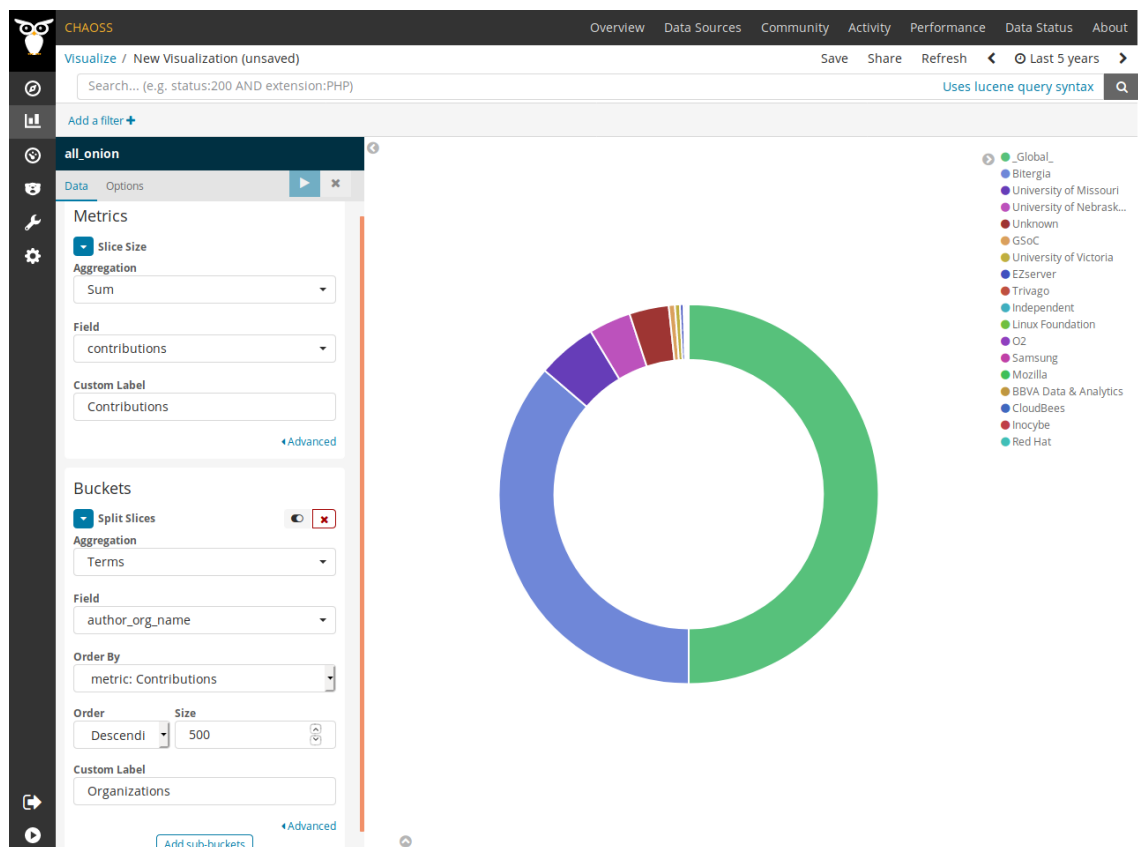
- 获取为项目做出贡献的组织列表。
- 查看各组织在特定时间内的贡献百分比。
- 查看组织结构在特定时间内的变化。
- 获取各组织相关的人员列表。

### 实现

- 从发生贡献的数据源收集数据。
- 确定贡献者的从属关系，对其从属组织作出准确评估。
- 关联贡献相关信息，并将其匹配至适当组织。
- 根据项目需求，您可能需要考虑：如何处理多个电子邮件地址、随时间变化的从属关系，承包商与员工的关系等问题。

### 提供指标的工具

- [GrimoireLab](#) 支持组织多元化指标，开箱即用。[GrimoireLab SortingHat](#) 管理身份信息。[GrimoireLab Hatstall](#) 用户界面可以修正人员的组织从属关系，甚至记录从属关系的变化。
  - 查看 [Bitergia Analytics](#) 的 [CHAOSS](#) 实例示例可视化效果。
  - 从 [GrimoireLab Sigils](#) 面板集合下载并导入包含此指标可视化效果示例的现成仪表板。
  - 按照说明向任意 [GrimoireLab Kibiter](#) 仪表板添加一个示例可视化效果：
    - 新建一个饼状图
      - 选择 `all_onion` 索引
      - 指标切片大小： `Sum` 聚合， `contributions` 字段， `Contributions` 自定义标签
      - 桶拆分切片： `Terms` 聚合， `author_or_name` 字段， `metric: Contributions` 排序依据， `Descending` 排列， `500` 大小， `Organization` 自定义标签
    - 屏幕截图示例



- **LF Analytics** 在主视图中为提交、提交的问题和通信渠道（目前支持 Slack 和 groups.io）提供组织多元化指标。

## Code

## Highlights



## Contributors

COMPANY   INDIVIDUAL			
	NAME	COMMITTS	%
1	IBM	444	332 72%
2	ING Bank	136	79 17%
3	(Robots)	102	51 11%

## 数据收集策略

## 定性

- 组织在项目或生态系统中的足迹
- 组织在项目或生态系统中的影响
- 治理结构中的从属多元化。

## 定量

- 每个组织的 commits 百分比



- 每个组织的 merges/reviews 百分比
- 每个组织的任意种类的贡献者百分比
- 每个组织贡献的代码行数百分比
- 每个组织提出的问题 (issues) 百分比
- [贡献组织](#) - 贡献组织的数量是多少?
- [新的贡献组织](#) - 新的贡献组织的数量是多少?
- 新贡献者组织 - 随着时间推移, 对项目作出贡献的新组织。
- 贡献组织的数量 - 在一段时间内参与项目的组织数量。
- 大象系数 - 如果 50% 的社区成员受雇于同一家公司, 那就形成了房间里的大象。公式: 员工完成 50% 提交次数的最小公司数量
- [从属多元化](#) - 单个公司的贡献者在所有贡献者中的比例。又称: 来自不同公司的维护者。贡献者从属多元化。
- 在具有代码所有权概念的项目中, 从属于每个组织的代码所有者的百分比, 按其拥有的代码的重要性/大小/LoC 和共同所有者的数量进行权衡。

## 参考资料

- 潜在实现和参考资料:
  - [https://bitergia.gitlab.io/panel-collections/open\\_source\\_program\\_office/organizational-diversity.html](https://bitergia.gitlab.io/panel-collections/open_source_program_office/organizational-diversity.html)
  - [Kata Containers 仪表板条目页面](#) (底部)
  - [Augur](#)

## Value WG

关注领域	目标
公共价值	确定项目 (包括下游项目) 是否对社区用户或者贡献者有价值。
Placeholder	Placeholder
人力投资	从组织的角度看项目是否具有经济价值。

## 关注领域 - 公共价值

**目标:** 确定项目 (包括下游项目) 是否对社区用户或者贡献者有价值。

度量指标	问题
项目发展速度	如何衡量组织的发展速度?

## 项目发展速度

问题：如何衡量组织的发展速度？

### 描述

项目的发展速度是指议题数量、代码提交数量、更改请求数量和贡献者个数，作为“创新”的指标。

### 目标

给开源项目办公室 (OSPO) 经理提供一种通过比较项目组合去衡量项目发展速度的方法。

OSPO 经理可以使用以下项目发展速度指标来：

- 比较开源项目与内部项目的发展速度
- 比较项目组合中的项目发展速度
- 找出哪些项目外部贡献者的发展超出了内部贡献者（在筛选内部贡献者与外部贡献者时）
- 找出值得投入的有前途的领域
- 找出未来几年可能成功的领域

参见示例

### 实现

基本指标包括：

- 关闭的问题个数
- 评论个数
- 修改代码的人数
- 代码提交人数

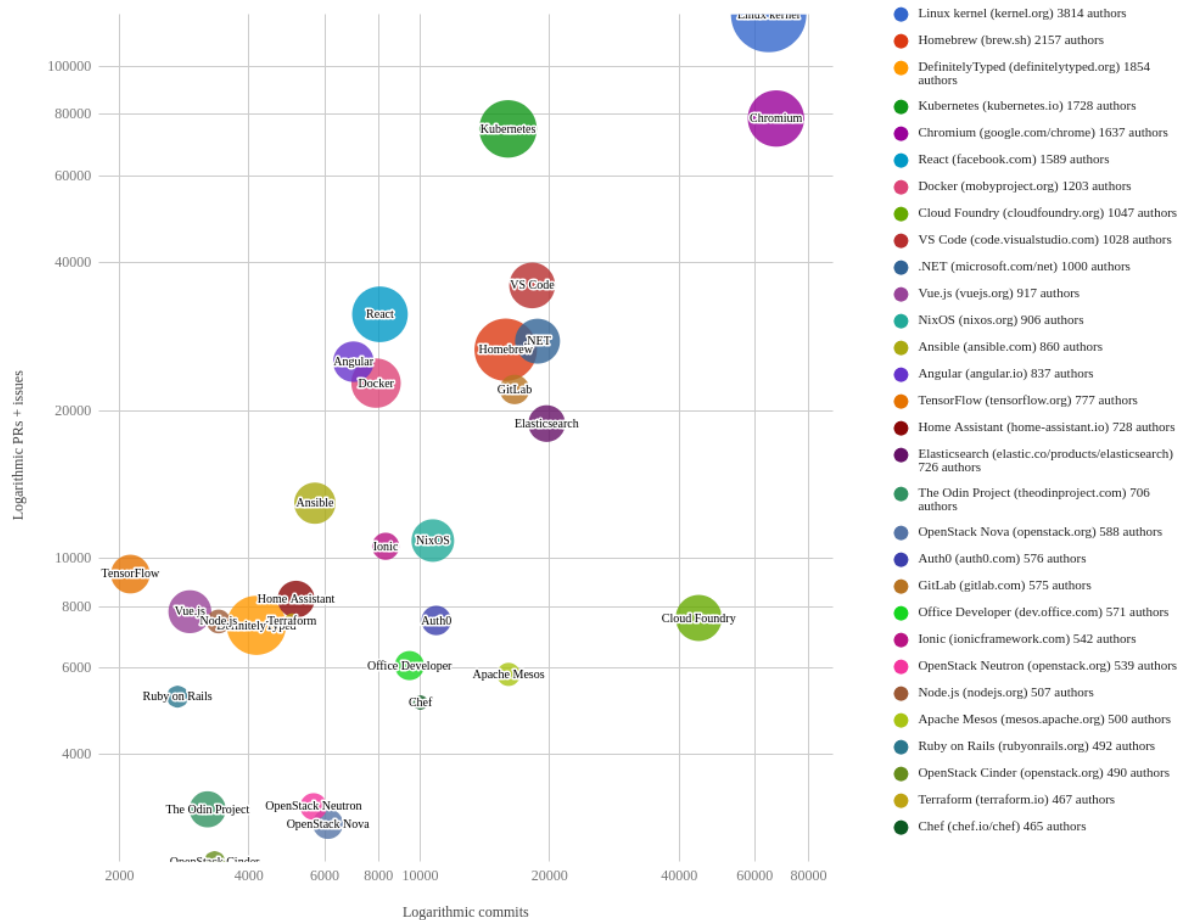
### 筛选条件

- 内部与外部贡献者
- 项目来源（例如，内部仓库、开源仓库和竞争对手的开源仓库）
- 时间

### 可视化效果

- X 轴：代码提交的总次数的对数
- Y 轴：提交问题总个数和评论总个数之和的对数。
- 点大小：代码提交人数
- 点为某个项目

Top 30 Projects 05/2016 - 04/2017



来自 CNCF

## 提供指标的工具

- CNCF - <https://github.com/cncf/velocity>

## 参考资料

- 开源创新能在企业中发挥作用吗?
- 高绩效文化的开放式创新
- 数字企业的开源
- 发展最快的开源项目

# 关注领域 - Placeholder

目标: Placeholder

度量指标	问题
就业机会 组织项目技能需求	有多少招聘信息要求项目技术与技能? 有多少组织在使用这个项目? 如果我精通这个项目, 是否可以雇用我?

## 就业机会

问题：有多少招聘信息要求项目技术与技能？

### 描述

开源贡献者赚取生活费的一种常见方式是受雇于公司或成为自雇或自由开发者。特定项目中的技能可能会改善求职者获得工作的前景。与特定开源项目中学技能相关的最明显需求指标是该项目或其技术纳入招聘信息中的时间。

### 目标

该指标可以让贡献者了解在特定开源项目中学技能被公司重视的程度。

### 实现

要在求职平台（如 LinkedIn、Indeed 或 Dice）上获得这一指标，前往职位搜索并输入开源项目的名称。返回的招聘信息数量即为指标。通过求职平台的 API 定期收集指标并存储结果，即可查看趋势。

### 筛选条件

- 招聘信息时长；信息会过时，填写后可能不会被移除

### 可视化效果

可以查看以下内容扩展指标：

- 返回的职位的薪资范围
- 返回的职位的资历水平
- 现场或非现场职位的可用性
- 工作地点
- 地理位置

### 参考资料

- LinkedIn 职位搜索 API： <https://developer.linkedin.com/docs/v1/jobs/job-search-api#>
- Indeed 职位搜索 API： <https://opensource.indeedeng.io/api-documentation/docs/job-search/>
- Dice.com 职位搜索 API： <http://www.dice.com/external/content/documentation/api.html>
- Monster 职位搜索 API： <https://partner.monster.com/job-search>
- Ziprecruiter API（需要合作）： <https://www.ziprecruiter.com/zipsearch>

注意：此指标仅限于单个项目，但参与开源也可由其他原因获益。这一指标可以超越单一项目，改为使用编程语言、进程、开源经验或框架等相关技能作为职位的搜索参数。

## 组织项目技能需求

问题：有多少组织在使用这个项目？如果我精通这个项目，是否可以雇用我？

### 描述

各组织通过用例和依赖关系来参与开源项目。该指标旨在确定与开源项目相关的下游技能需求。该指标着眼于将项目作为 IT 基础设施的一部分进行部署的组织、其他具有声明依赖关系的开源项目，以及通过社交媒体、会议提及、博客文章和类似活动对项目的引用。

### 目标

作为一名开发者，我希望把自己的技术和时间投入到有可能让我在未来获得一份体面薪水的项目中。人们可以使用项目软件的下游组织影响指标来了解组织使用的项目，由此寻求可能需要 IT 支持服务的工作机会。

### 实现

基本指标包括：

- 为项目创建问题的组织数量
- 为项目创建拉取请求的组织数量
- 发布项目相关博客或推文的组织数量
- 在公开招聘信息中提到项目的组织数量
- 派代表参加项目相关会议的组织数量
- 依赖某项目的其他项目数
- 项目相关书籍的数量
- 项目的 Google 搜索趋势

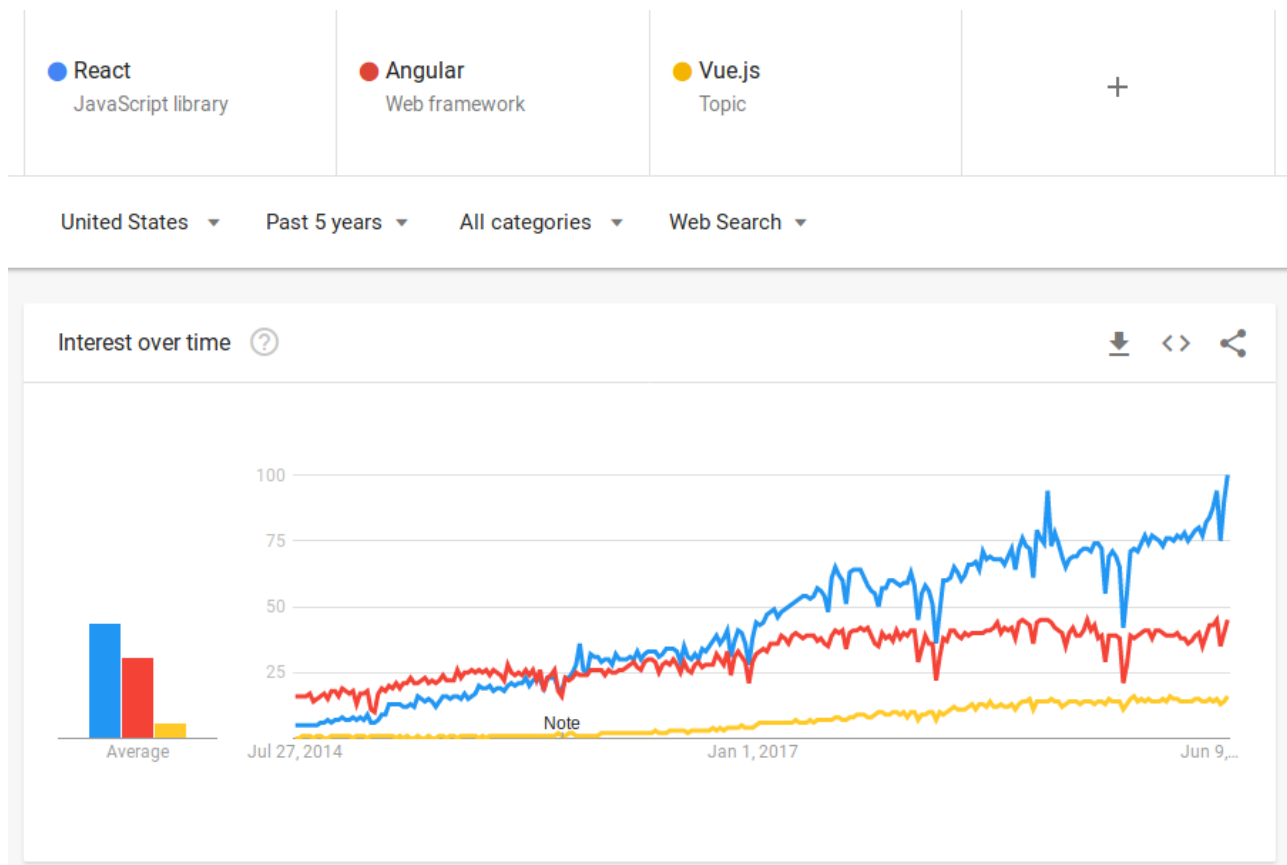
### 可视化效果

以下可视化效果展示了依赖于特定项目的下游项目的数量。虽然这种可视化效果未捕捉项目软件下游组织影响指标的整体，但它提供了一部分的可视化效果。

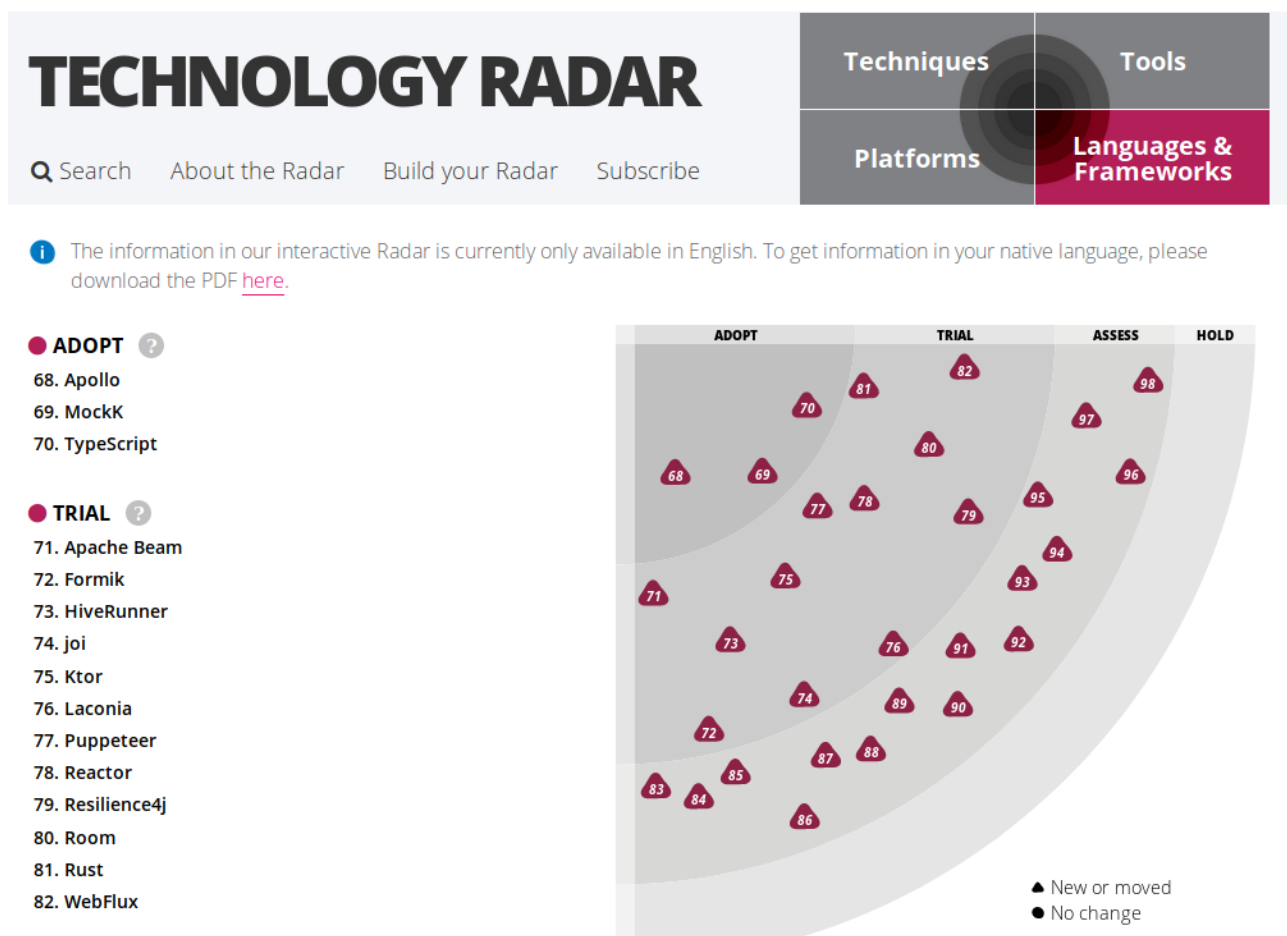




其他可视化效果可能包括 Google 搜索趋势（React vs. Angular vs. Vue.js）。



ThoughtWorks 发布了名为'Tech Radar' 的系列，展现技术的普及程度。



Tech Radar 让您能够对项目进行深入研究，了解评估随时间的变化。

## React.js

NOV  
2016

### ADOPT ?

In the avalanche of front-end JavaScript frameworks, [React.js](#) stands out due to its design around a reactive data flow. Allowing only one-way data binding greatly simplifies the rendering logic and avoids many of the issues that commonly plague applications written with other frameworks. We're seeing the benefits of React.js on a growing number of projects, large and small, while at the same time we continue to be concerned about the state and the future of other popular frameworks like [AngularJS](#). This has led to React.js becoming our default choice for JavaScript frameworks.

APR  
2016

### ADOPT ?

NOV  
2015

### TRIAL ?

One benefit of the ongoing avalanche of front-end JavaScript frameworks is that occasionally a new idea crops up that makes us think. [React.js](#) is a UI/view framework in which JavaScript functions generate HTML in a reactive data flow. It differs significantly from frameworks like [AngularJS](#) in that it only allows one-way data bindings, greatly simplifying the rendering logic. We have seen several smaller projects achieve success with React.js, and developers are drawn to its clean, composable approach to componentization.

MAY  
2015

### TRIAL ?

One benefit to the ongoing avalanche of front-end JavaScript frameworks is that occasionally, a new idea crops up that makes us think. [React.js](#) is a UI/View framework in which JavaScript functions generate HTML in a reactive data flow. We have seen several smaller projects achieve success with React.js and developers are drawn to its clean, composable approach to componentization.

### NOT ON THE CURRENT EDITION

This blip is not on the current edition of the radar. If it was on one of the last few editions it is likely that it is still relevant. If the blip is older it might no longer be relevant and our assessment might be different today. Unfortunately, we simply don't have the bandwidth to continuously review blips from previous editions of the radar

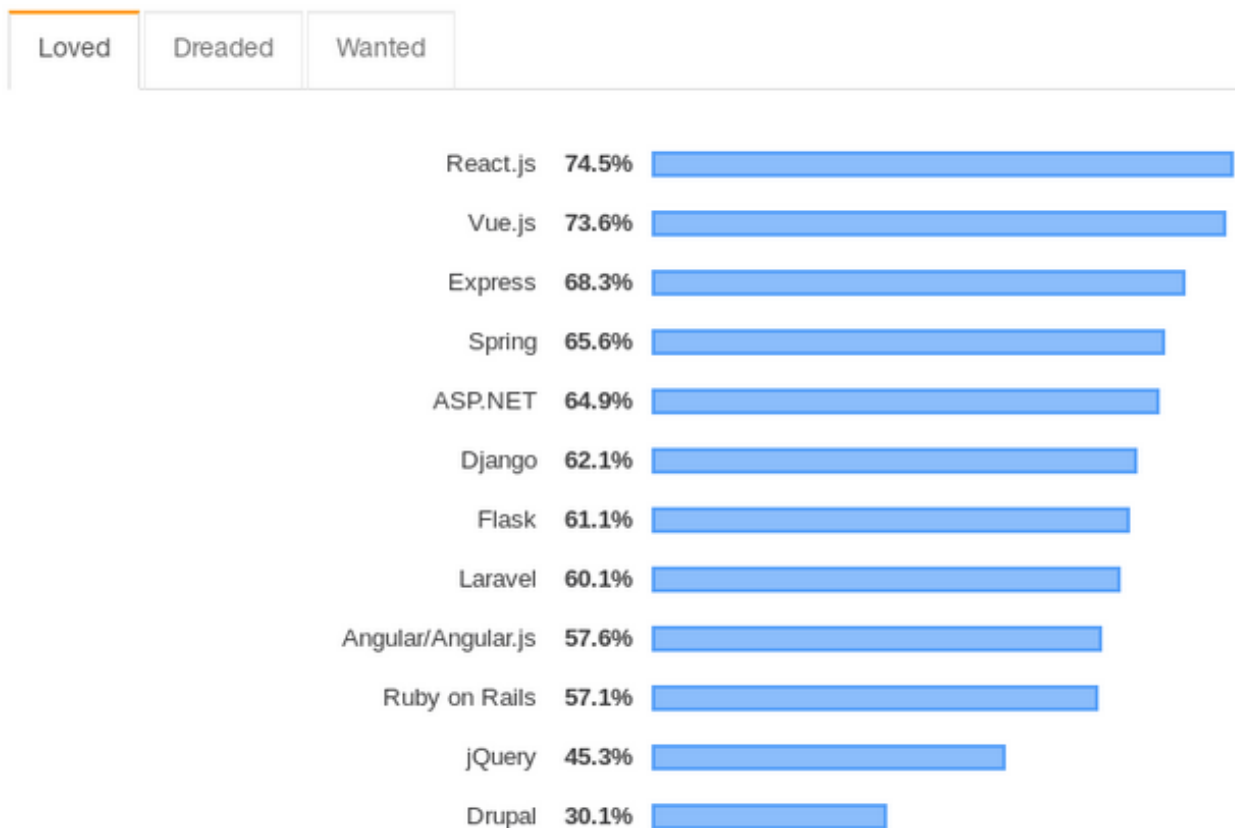
[Understand more »](#)

StackOverview 发布了年度开发者调查



## Most Popular Technologies

### Most Loved, Dreaded, and Wanted Web Frameworks



% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

[React.js](#) and [Vue.js](#) are both the most loved and most wanted web frameworks by developers, while [Drupal](#) and [jQuery](#) are most dreaded.

### 提供指标的工具

- Google 趋势 - 显示一段时间内的搜索兴趣
- ThoughtWorks TechRadar - 技术咨询公司的项目评估
- StackOverflow 开发者调查 - 年度项目排名
- Augur; 示例可用于多个仓库：
  - [Rails](#)
  - [Zephyr](#)
  - [CloudStack](#)

### 参考资料

- 开源赞助
- 财政赞助和开源
- 大型企业开源赞助
- Google 趋势 API
- 衡量开源软件的影响
- ThoughtWorks Tech Radar
- Stack Overflow 开发者调查

## 关注领域 - 人力投资

**目标:** 从组织的角度看项目是否具有经济价值。

度量指标	问题
人力投资	组织投入人力对社区所做的贡献（例如：代码提交，议题和更改请求）花费的成本是多少

## 人力投资

问题：组织投入人力对社区所做的贡献（例如：代码提交，议题和更改请求）花费的成本是多少

### 描述

开源项目通常由组织的人力投入来支撑。该指标跟踪组织对单个项目的经济投入（体现在人力成本）。

### 目标

随着组织参与度对开源项目变得越来越重要，组织必须清楚了解其人力投资。该指标的目的在于为从事开源项目的组织提高人力成本的透明度。该指标给开源项目办公室（OSPO）经理提供了一种通过项目投资组合比较人力成本的方法。比如：人力投资指标能用在确定投资的优先顺序或者确定投资回报。例如：

- 以人力投资评估 OSPO 事务的优先级和证明预算合理性
- 以人力投资解释产品、项目管理事项的优先级
- 以人力投资论证继续投资 OSPO 的价值
- 以人力投资反应和比较开源贡献与内部工作的人力成本
- 以人力投资比较项目组合的项目效益

### 实现

基础指标包括：

- 贡献数量
- 按贡献者类型（内部/外部）划分的贡献数量
- 按贡献类型（如代码提交，议题和更改请求）划分的贡献数量

参数包括：

- 每小时劳动率
- 创建贡献的平均劳动时间（按照贡献类型分类）

人力投资 = 每一种贡献类型的总和（贡献数量 \* 创造贡献的平均工时 \* 平均每小时劳动率）

### 筛选条件

- 内部与外部贡献者
- 问题标签
- 项目来源（如内部、开源仓库、竞争对手的开源仓库）

### 可视化效果

1	IssueID	Severity	Title	Status	Contributor	Tag
2	34234	High	Add CSV Graphic	Open	andyl	metrics
3	23421	Med	Fix typos	Closed	mattg	metrics
4	56743	High	Reword section	Open	georg	augur
5	85879	Low	Add CNCF PNG	Open	seang	metrics
6	34183	High	Remove button	Closed	vinod	implementation
7	76790	Low	Use large font	Open	kevin	metrics
8	57432	Med	Sync with web	Closed	carol	implementation

我们的第一个参数化指标的可视化效果依赖于可以用 Augur 导出的 CSV。电子表格用于指标参数和计算公式。未来的实现可能会在 webapp 中直接添加参数操作的功能。

## 参考资料

- [启动开源项目办公室](#)
- [创办开源项目办公室](#)
- [企业开源](#)



## Release History

---

CHAOSS metrics are released continuously. The regular release is when we update the version number, update the full release notes, and make a big announcement. These releases occur one to two times a year and may correspond with the dates of a CHAOSScon event. Prior to regular release, continuous released metrics go through a comment period of at least 30 days.

## Continuous Metric Contributions Since Last Release

---

- **Common WG**
  - New metrics include:
  - Name Change/Revision:
- **Diversity & Inclusion WG**
  - New metrics include:
  - Name Change/Revision:
- **Evolution WG**
  - New metrics include:
  - Name Change/Revision:
- **Risk WG**
  - New metrics include:
  - Name Change/Revision:
- **Value WG**
  - New metrics include:
  - Name Change/Revision:

## Release 2021-03 Notes:

---

- [PDF of released CHAOSS Metrics \(v.2021-03\)](#)
- **Common WG**
  - New metrics include:
    - Technical fork
    - Burstiness
    - Review Cycle Duration within a Change Request
- **Diversity & Inclusion WG**
  - New metrics include:
    - Chat Platform Inclusivity
    - Documentation Accessibility
    - Project Burnout
- **Evolution WG**
  - New metrics include:
    - Branch lifecycle
    - Change Request Acceptance ratio
  - Name Change/Revision
    - Change Requests accepted
    - Change Requests declined
    - Change Requests Duration
    - Change Requests
- **Risk WG**
  - New metrics include:
    - SPDX Document
    - Bus Factor
- **Value WG**
  - New metrics include:
    - Project popularity
  - Name Change/Revision
    - Social Listening

## Release 2020-08 Notes:

---

- [PDF of released CHAOSS Metrics \(v.2020-08\)](#)
- **Common WG**
  - New metrics include:
    - Contributor Location
    - Time to Close
    - Types of Contributions
- **Diversity & Inclusion WG**
  - New metrics include:
    - Documentation Usability
    - Inclusive Leadership
    - Issue Label Inclusivity
  - New focus area - Project and Community
- **Evolution WG**
  - New metrics include:
    - Inactive Contributors
    - New Contributors
  - The Reviews metric was revised
- **Risk WG**
  - No new metrics this release
  - The Elephant Factor metric was revised
- **Value WG**
  - No new metrics this release
  - The SCMS metric was revised
  - Work group focused on restructuring and creation of new focus areas - Organizational Value, Individual Value, and Communal Value
  - All previously released metrics were assigned to the new focus areas

## Release 2020-01 Notes:

---

- [PDF of released CHAOSS Metrics \(v.2020-01\)](#)
- All Metrics were restructured to conform to the new CHAOSS Project metrics document structure.
- **Common WG**
  - New metrics include:
    - Activity Dates and Times
    - Time to First Response
    - Contributors
  - Restructured and renamed focus areas
  - Organizational Diversity remains unchanged from previous release.
- **Diversity & Inclusion WG**
  - New metrics include:
    - Sponsorship
    - Board/Council Diversity
  - Improved clarity on several metrics that were in the previous release
- **Evolution WG**
  - New metrics include:
    - Issue Age
    - Issue Response Time
    - Issue Resolution Duration
    - New Contributors Closing Issues
  - Updated focus areas. Refactored the "Code Development" focus area into 3 separate focus areas to more closely align with other working groups. Rather than having one broad focus area with multiple subsections, we decided our intent would be better communicated by making each of these subsections into their own focus areas. The 3 separate focus areas include:
    - Code Development Activity
    - Code Development Efficiency
    - Code Development Process Quality
  - Kept the other 2 focus areas (Issue Resolution and Community Growth) the same.
  - No major changes were made to already existing metrics.
- **Risk WG**
  - New metrics include:
    - OSI Approved Licenses
    - Licenses Declared
    - Test Coverage (Updated)
    - Elephant Factor
    - Committers
  - Focused on increasing metrics coverage in the general areas of CNCF Core Infrastructure badging and licensing.
  - License Count was removed from the release. It is being replaced by the Licenses Declared metric.
  - Software Bill of Materials was removed from the release. It is being reworked as the SPDX Document metric for the next release.
- **Value WG**
  - New metrics include:
    - Social Currency Metric System (SCMS)
    - Job Opportunities
  - A new focus area of Ecosystem Value was developed

## Release 2019-08 Notes

---

- [PDF of released CHAOSS Metrics \(v.2019-08\)](#)
- Initial CHAOSS Metrics release.

# LICENSE

---

## MIT License

Copyright (c) 2021 CHAOSS

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.