



Metrics

Release 2021-03

<https://chaoss.community/metrics>

MIT License

Copyright © 2021 CHAOSS a Linux Foundation® Project

Contents

CHAOSS Contributors	2
CHAOSS Governing Board Members	3
Value WG	4
Focus Area - Communal Value	5
Project Popularity	6
Project Velocity	9
Social Listening	11
Focus Area - Individual Value	17
Job Opportunities	18
Organizational Project Skill Demand	19
Focus Area - Organizational Value	25
Labor Investment	26
Release History	28
Continuous Metric Contributions Since Last Release	29
Release 2021-03 Notes:	30
Release 2020-08 Notes:	31
Release 2020-01 Notes:	32
Release 2019-08 Notes	33
LICENSE	34

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 Steinhacher, 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, Pranjali 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

Value WG

Focus Area	Goal
Communal Value	Identify if a project is valuable to its community of users (including downstream projects) or contributors.
Individual Value	Identify if a project is valuable to me as an individual user or contributor.
Organizational Value	Identify if a project is monetarily valuable from an organization's perspective.

Focus Area - Communal Value

Goal: Identify if a project is valuable to its community of users (including downstream projects) or contributors.

Metric	Question
Project Popularity	How popular is an open source project?
Project Velocity	What is the development speed for an organization?
Social Listening	How does one measure the value of community interactions and accurately gauge “reputation” of a community as evident from qualitative sentiment?

Project Popularity

Question: How popular is an open source project?

Description

Project popularity can be measured by how much activity is visible around a project. Popularity has a positive feedback loop in which more popular projects get more attention, attract more users or developers, and see increases in popularity, spinning the popularity wheel.

Project popularity may be used as a proxy for understanding project value because open source project economic value is hard to measure, due to a lack of available usage or sales information for open source projects.

Objectives

In a quest to earn a living wage, and to maximize future employment opportunities, workers may be interested in knowing which projects are growing and are underserved. Similarly, from an organizational perspective, knowing which projects are highly used can be helpful in knowing which projects might be worth investing in. The Project Popularity metric can be used to identify the trajectory of a project's development.

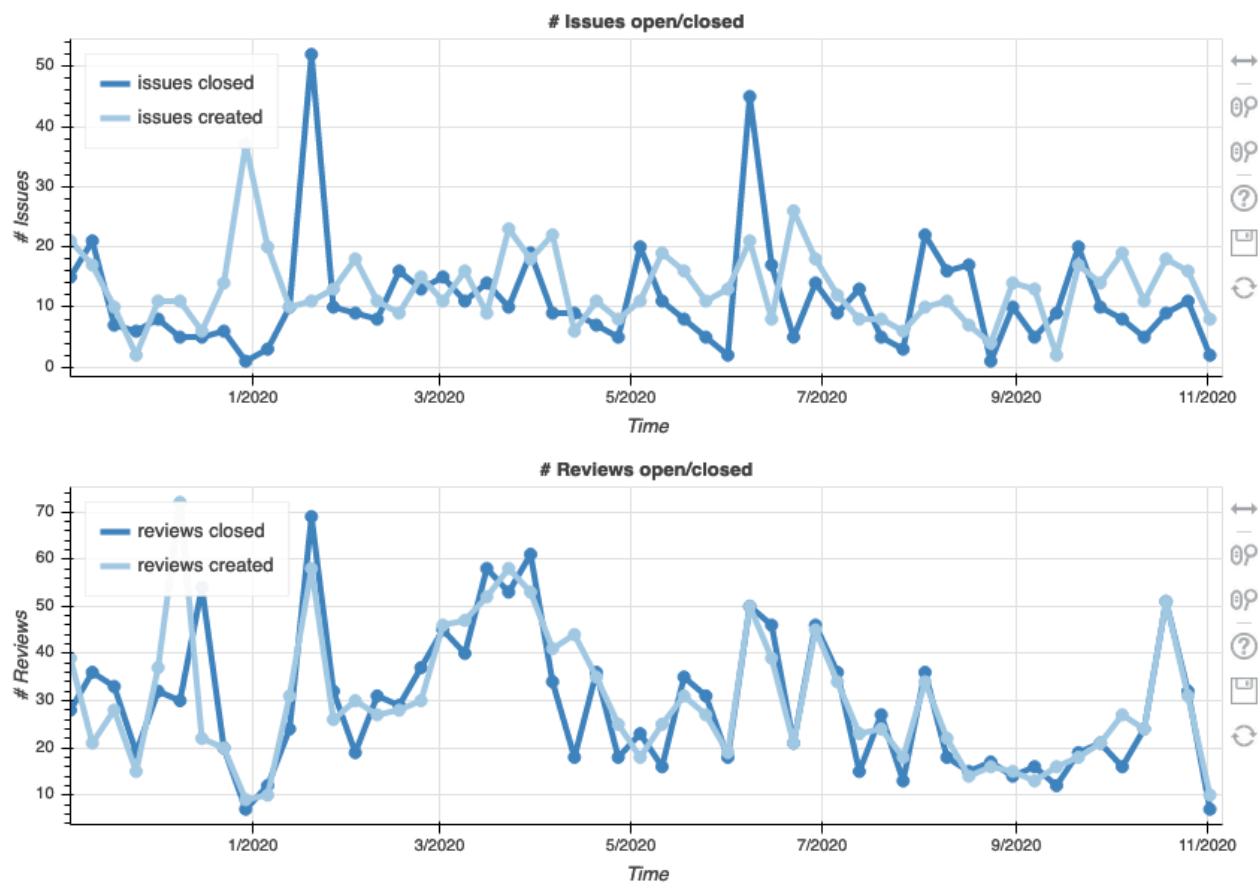
Implementation

The project popularity metric is often considered with changes over time. There are numerous example vectors to consider when measuring project popularity based on the number of:

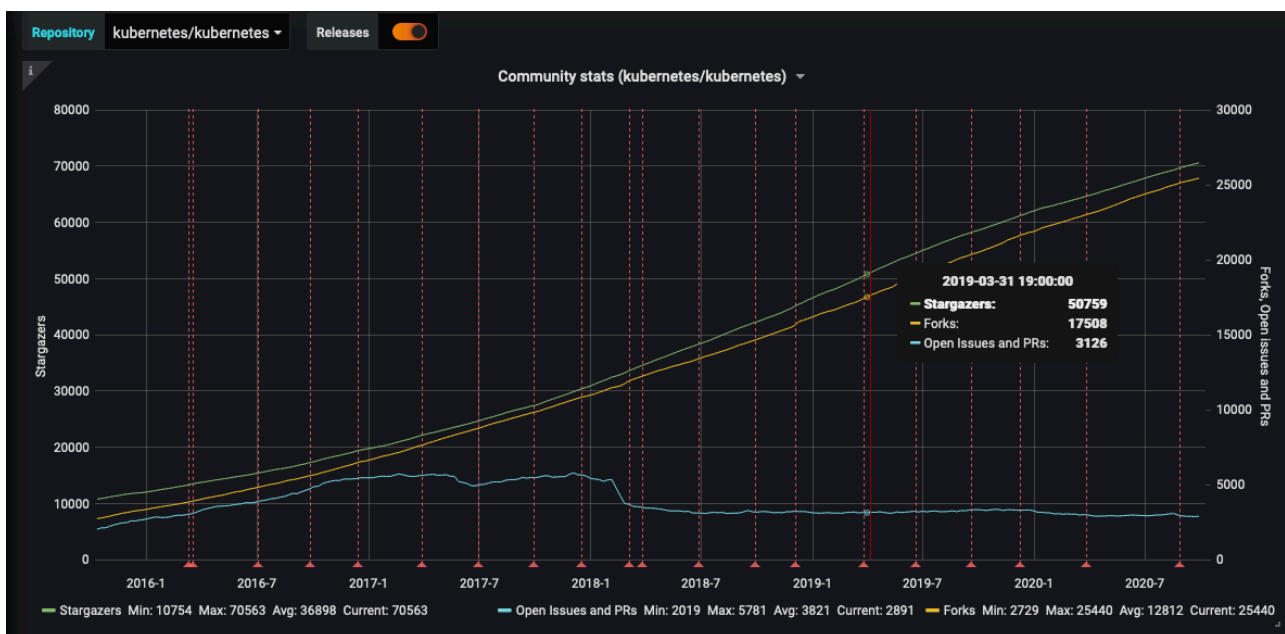
1. Social media mentions
2. Forks
3. [Change requests](#)
4. [New Issues](#)
5. Stars, badges, likes
6. [New contributors](#)
7. [Organizational Diversity](#)
8. Job postings requesting skills in project
9. Conversations within and outside of project
10. Clones
11. Followers
12. Downstream dependencies
13. People attending events that focus on a project

Visualizations

Issues and reviews (change requests) visualization from Cauldron (GrimoireLab):



Kubernetes project popularity statistics from DevStats:



Tools Providing the Metric

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

References

- [Popular OpenSource Projects](#)
- [Is It Maintained?](#)
- [Open Source Project Trends](#)
- [Kubernetes Salary](#)

Project Velocity

Question: What is the development speed for an organization?

Description

Project velocity is the number of issues, the number of pull requests, volume of commits, and number of contributors as an indicator of 'innovation'.

Objectives

Gives an Open Source Program Office (OSPO) manager a way to compare the project velocity across a portfolio of projects.

The OSPO manager can use the Project Velocity metric to:

- Report project velocity of open source projects vs in-house projects
- Compare project velocity across a portfolio of projects
- Identify which projects grow beyond internal contributors (when filtering internal vs. external contributors)
- Identify promising areas in which to get involved
- Highlight areas likely to be the successful platforms over the next several years

[See Example](#)

Implementation

Base metrics include:

- [issues closed](#)
- [number of reviews](#)
- [# of code changes](#)
- [# of committers](#)

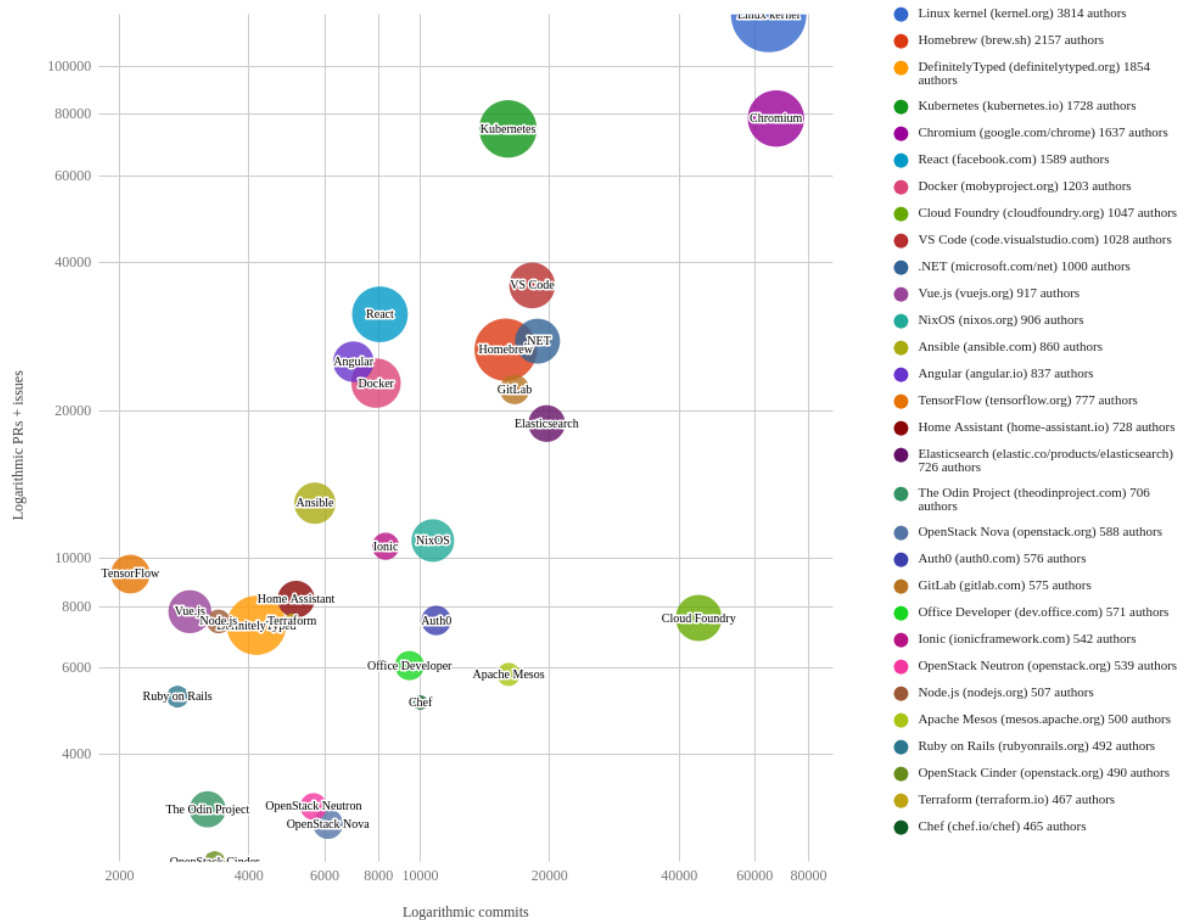
Filters

- Internal vs external contributors
- Project sources (e.g., internal repositories, open-source repositories, and competitor open-source repositories)
- Time

Visualizations

- X-Axis: Logarithmic scale for Code Changes
- Y-Axis: Logarithmic scale of Sum of Number of Issues and Number of Reviews
- Dot-size: Committers
- Dots are projects

Top 30 Projects 05/2016 - 04/2017



From CNCF

Tools providing the Metric

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

References

- Can Open Source Innovation work in the Enterprise?
- Open Innovation for a High Performance Culture
- Open Source for the Digital Enterprise
- Highest Velocity Open Source Projects

Social Listening

Question: How does one measure the value of community interactions and accurately gauge “reputation” of a community as evident from qualitative sentiment?

Note: This metric synthesizes several other metrics that can be derived from trace data, and several process-oriented metrics. Embedded footnotes annotate areas planned for later clarification, and questions for later resolution.

Description

Social Listening is a combination of [social listening](#) practices across multiple channels along with a meaningful set of categorizations. The combination of these tactics can lead to systematic community analysis and can inform a community strategy that leads to measurable business value. [1]

Theory and Origin

Social currency or social capital is a social scientific theory. It broadly considers how human interactions build relationships and trust in a community. The Social Listening metric represents the reputation of a community as measured via community trust, transparency, utility, consistency, and merit.

Interpersonal relationships are the social fabric of communities. This is shown in the [Levinger's Relationship Model](#) and [Social Penetration Theory](#). Community members' sense of personal and group identity grows as they interact. Members build shared values, accumulate a sense of trust, encourage cooperation, and garner reciprocity through acts of [self-disclosure](#). These interactions build an increased and measurable sense of connection. The measure of these characteristics is called social currency. [2]

Results

The Social Listening metric is a way to sort through a fire hose of qualitative data from community interactions. A central premise of this approach is that community members' interactions have an impact on the community. The Social Listening metric continually measures the sentiment [3] from those interactions. It illustrates the reputation and level of trust between community members and leaders. [4]

Objectives

Analyze the qualitative comments in community interactions. Gain an overview of sentiment in a community. Get metrics that show at a glance how a community is and was doing. Use lead metrics from continuous measurements for proactive community strategy development. Instill trust in community members that their thoughts and opinions are valued.

Implementation

The Social Listening requires the collection of community comments (communication traces), the definition of a codex, and the on-going review of the communication traces. [5]

Set up a Data Collection Platform of your choice as described in the “Tools” section below. Ensure it has a minimum of 4 dimensions and 3 communication channels. Once it is set up, the following method is used to collect, analyze, and interpret results:



1. **Collect Communication Traces** -- Identify online platforms that your community is communicating on. Set up data funnels from the primary platform to your Social Listening tool. The critical data for the system is user generated content.
2. **Standardize How Communication Traces Should Be Assessed** -- Use a codex to define important concepts as a “tracking keyword” or “category” in the focal community. This unified codex of terms ensures consistent analysis as different people read and tag community sentiment. Formalizing the revision and addition structure to this codex on a regular basis is a must. [5]
3. **Analyze the Communication Traces** -- Community sentiment is analyzed in the Social Listening tool by tagging data with codex terms. If the tagging is done by a team of people, it is recommended that everyone gets together regularly to discuss trends and ensure consistent tag use. If the tagging is done by an artificial intelligence algorithm, then a human team should supervise and retrain the AI as necessary. [5]
4. **Share and Visualize the Aggregated Analysis** -- Visualize the quantitative count of codex terms over time, e.g., in a dashboard. This is where the qualitative analysis results produce an easy to observe dashboard of trends. Share analysis with team members. [6]
5. **Benchmark, Set Goals & Predict Future Growth** -- After getting enough data to form a benchmark, take stock of where your community stands. What are its strengths and weaknesses? What actions can be taken to make the community healthier and more robust? Then form community initiatives with well-defined goals and execute on these projects to affect the social currency metrics for next week. [6]
6. **Repeat the Process** -- In regular evaluation meetings, discuss the shortcomings of the dataset or collection methods. Come up with methods to address these shortcomings in the future. Work solutions into the system and move forward. Truth is in the trend, power is in the pattern. [7]

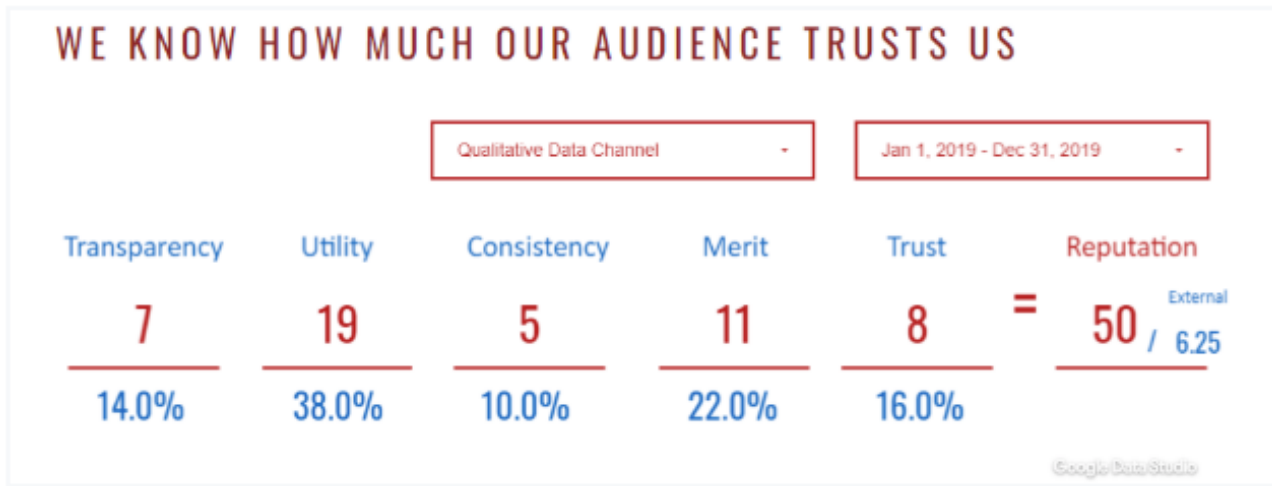
Filters

1. **Channel:** Sort by where the data was collected from.
2. **Tag:** Show data based on what codex tags were used to identify sentiment in comments.
3. **Time:** Show trends in the data over time and pull specific data-sets.
4. **Most impactful comments:** Sort and filter by flags that can be placed in the data to highlight specific data points and explain their importance.
5. **AI vs. Human tagged:** Filter by whether tags were applied programmatically or by a person.

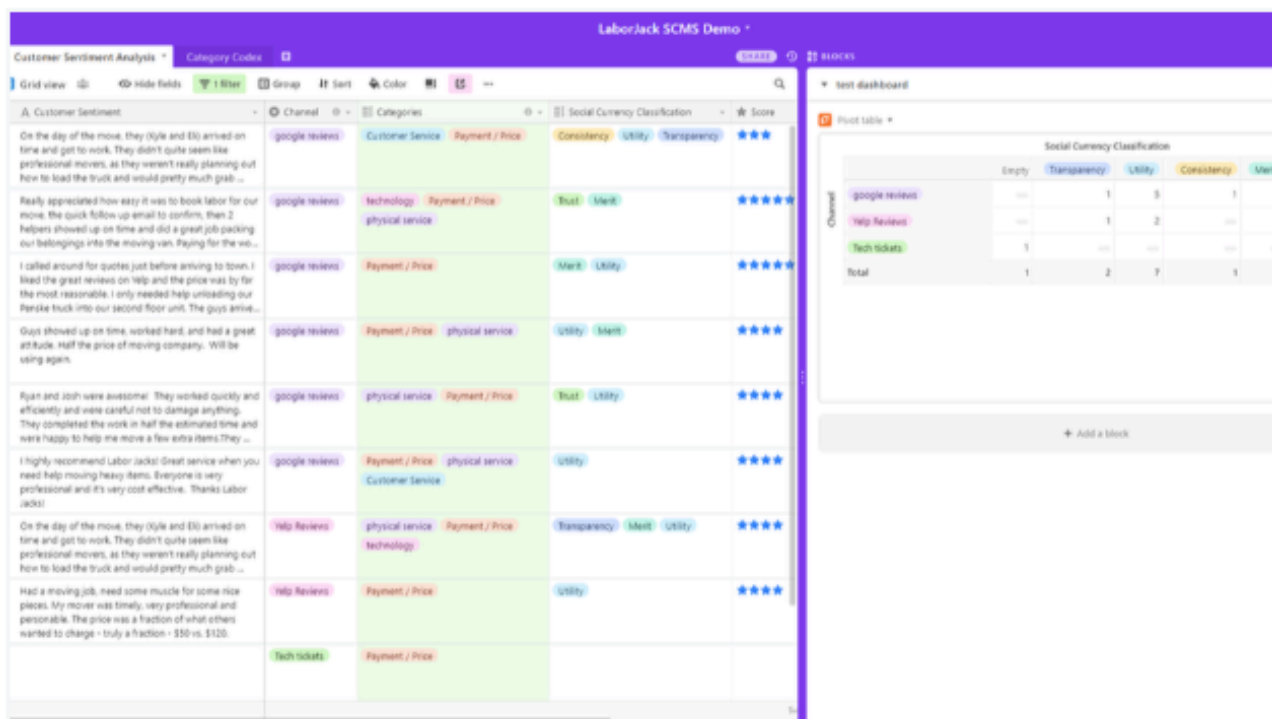
6. **Weighted currency:** Weight the “importance” of certain comments based on any one individually selected criteria. A resulting weighted view is simply a re-order of information based on weight.

Visualizations

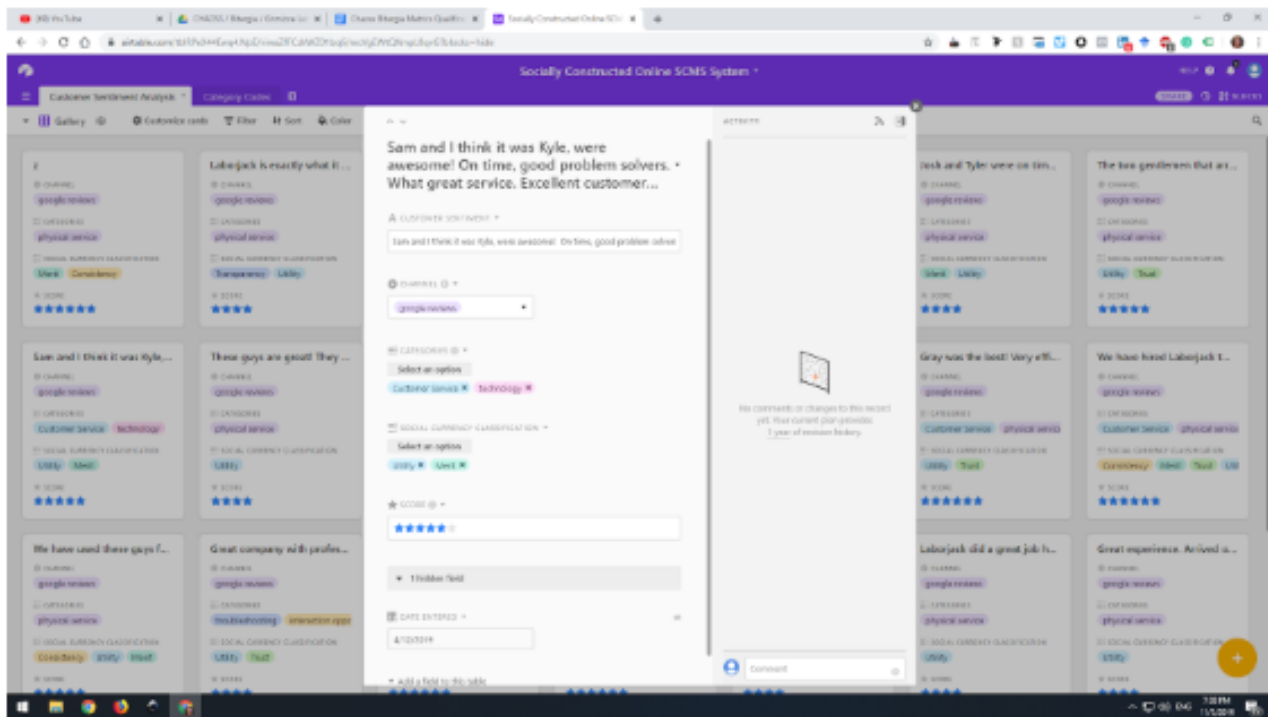
Dashboard visualizing the aggregate metrics:



Example Social Listening tool: On the left, raw community comments are shown and tags are added in columns immediately to the right. On the right, a pivot table shows in numbers how often tags occurred in combination with other tags.



Expanded comments view: remove the “quantitative” from the fields and provide the best possible way to read the different comments.



Tools Providing the Metric

To implement the metric any MySQL, smart-sheet, excel, or airtable-like excel datasheet program works fine. This data should be simplified enough to interact with other data interfaces to ensure that data migration is simple, straightforward, and can be automated (such as google data studio). This requires that systems used to implement the Social Listening metric work with CSV and other spreadsheet files, and we heavily recommend open source programs for its implementation.

Once you have this, create a data set with the following data points: [8]

Data Points	Description
Date of entry	Date data was imported to Social Listening tool
Date of comment	Date comment was made on original platform
Comment Text	Qualitative data brought in. Decide on how large you want these chunks ported. Some may port an entire email while others will be broken into one row per sentence. It should only have one "sentiment"
Data channel	Originating data channel the comment came from
Tags (created on codex document below)	Based on the unified codex of terms, decide what tags to track. There can be two kinds of tags. On the one hand, tags can be based on "themes" or recurring sentiment that people voice (e.g., gamer gate, flamewar, or thank you notes). On the other hand, tags based on "categories" can describe different aspects of a community that members comment on (e.g., events, release, or governance).
Social Currency Metric	The social currency being awarded or demerited in the system. This will directly affect numbers.
Weighted Score	Once you've decided what your "weight" will be, you can assign a system of -3 to +3 to provide a weighted view of human-tagged metrics (AI will not assign a weight for several reasons). This enables the "most impactful comment" filter.

Create a second sheet for the Unified Codex of Terms which will define terms. It should look like this: [8]

Category Term	Definition	When to use	When not to use
[Custom Tags - themes and categories]			
[Community specific jargon]			
Social Currency Dimensions:			
TRANSPARENCY	Do people recognize and feel a connection to your community?	When they have the "words" to pinpoint why they feel you are authentic or personable.	This is not about good customer service, or doing well. That is utility. This is about whether they understand who you are as a business and show they are onboard with it.
UTILITY	Is your community doing something useful or is it contributing value?	Provide parameters that exclude when the term is used so that people know when the category tag should not be implemented.	This is not about good customer service, or doing well. That is utility. This is about whether they understand who you are as a business and show they are onboard with it.
CONSISTENCY	Do you have a history of being reliable and dependable?	When they suggest they have used your brand, or interacted with you several times	If they've only provided their comment to suggest you were useful once, use utility instead.
MERIT	Does your community merit respect and attention for your accomplishments?	When the social currency garnered from customers seems it will continue for a while, and will impact other people's opinions.	When they suggest they will use you again in the future use trust instead as that is a personal trust in the brand. Merit is external.
TRUST	Can people trust that your community will continue to provide value and grow in the future?	When they suggest they trust you well enough to continue conversations with you in the future	When there is not substantial enough evidence to suggest they will continue to work with and trust you as a loyal customer or community member.
INTERNAL REPUTATION ⁹	Do people believe these things strongly enough to warrant conversation or action?		
EXTERNAL REPUTATION ⁹	What amount of your reputation in your community is transferable to strangers outside of your community (cold audiences)?		

The codex is filled in by stakeholders on a regular basis by specific communities and forms the basis for analysis of the data. This is the MOST IMPORTANT part. Without this the subjectivity of qualitative data does not follow the rule of generalization: [9]

"A concept applies to B population ONLY SO FAR AS C limitation."

Data Collection Strategies

Community member comments are available from trace data. The Social Listening metric ideally imports the comment text automatically into a tool for tagging. Trace data can be collected from a communities' collaboration platforms where members write comments, including ticketing systems, code reviews, email lists, instant messaging, social media, and fora.

Legal and Regulatory Considerations

Points of destruction: Detailed data is destroyed after xx months has passed. Quantitative calculations are kept for up to 250 weeks per GDPR compliance. Data older than 250 weeks becomes archived data you cannot

manipulate but can see. Users can negotiate the primary statistic.

References

- [An example implementation on airtable](#)
- [An example implementation in data studio \(report\)](#)
- [An example implementation in data studio \(data source\)](#)
- [An example implementation in google sheets](#)
- [Implementation documentation](#) (starts on page 33)

Annotated Footnotes

[1] CHAOSS metrics historically is to create standard definitions that can be used reliably across projects to support comparisons. This metric may evolve into a project in the future.

[2] What metrics emerge from this description? Likely included are: 1. community trust, 2. transparency, 3. utility, 4. consistency, and 5. merit

[3] Analysis of sentiment suggests that metric (6) is likely "Communications Sentiment", and the definition may need to include references to common sentiment analysis tools, sometimes called "bags of words".

[4] Measuring how trust is instilled in community members, such that their thoughts and opinions are valued is likely metric (7) that will define a process, and perhaps is not measurable via trace data.

[5] A substantial portion of any codex for open source software will be common across projects, and each project is likely to have a set of particular interests that are a subset of that codex. In some cases, their main interests may not be present in an established codex component. In general, the codex, like the CHAOSS project itself, is open sourced as shared metadata to ensure shared understanding across open source communities.

[6] This describes the evolution of a standard codex, and its elements through the process of CHAOSS working groups and projects, characterized in the previous footnote. Likely this will be a process metric (8).

[7] Candidate process oriented metric (9).

[8] Examples of data coded using the open sourced codex, as it evolves, will be essential components for advancing open source software through Social Listening. Implementations will require these examples, and their provision as open source assets of the CHAOSS project will return value as shared data.

[9] Internal and external reputation are likely metrics (10), and (11) arising from the Social Listening metric.

Focus Area - Individual Value

Goal: Identify if a project is valuable to me as an individual user or contributor.

Metric	Question
Job Opportunities	How many job postings request skills with technologies from a project?
Organizational Project Skill Demand	How many organizations are using this project and could hire me if I become proficient?

Job Opportunities

Question: How many job postings request skills with technologies from a project?

Description

A common way for open source contributors to earn a living wage is to be employed by a company or be a self-employed or freelance developer. Skills in a specific project may improve a job applicant's prospects of getting a job. The most obvious indicator for demand related to a skill learned in a specific open source project is when that project or its technology is included in job postings.

Objectives

The metric gives contributors a sense of how much skills learned in a specific open source project are valued by companies.

Implementation

To obtain this metric on a job search platform (e.g., LinkedIn, Indeed, or Dice), go to the job search and type in the name of the open source project. The number of returned job postings is the metric. Periodically collecting the metric through an API of a job search platform and storing the results allows to see trends.

Filters

- Age of job posting; postings get stale and may not be removed when filled

Visualizations

The metric can be extended by looking at:

- Salary ranges for jobs returned
- Level of seniority for jobs returned
- Availability of jobs like on-site or off-site
- Location of job
- Geography

References

- LinkedIn Job Search API: <https://developer.linkedin.com/docs/v1/jobs/job-search-api#>
- Indeed Job Search API: <https://opensource.indeedeng.io/api-documentation/docs/job-search/>
- Dice.com Job Search API: <http://www.dice.com/external/content/documentation/api.html>
- Monster Job Search API: <https://partner.monster.com/job-search>
- Ziprecruiter API (Requires Partnership): <https://www.ziprecruiter.com/zipsearch>

Note: This metric is limited to individual projects but engagement in open source can be beneficial for other reasons. This metric could be tweaked to look beyond a single project and instead use related skills such as programming languages, processes, open source experience, or frameworks as search parameters for jobs.

Organizational Project Skill Demand

Question: How many organizations are using this project and could hire me if I become proficient?

Description

Organizations engage with open source projects through use and dependencies. This metric is aimed at determining downstream demand of skills related to an open source project. This metric looks at organizations that deploy a project as part of an IT infrastructure, other open source projects with declared dependencies, and references to the project through social media, conference mentions, blog posts, and similar activities.

Objectives

As a developer, I'd like to invest my skills and time in a project that has a likelihood of getting me a decent paying job in the future. People can use the Downstream Organizational Impact of a Project Software metric to discover which projects are used by organizations, and they may, therefore, be able to pursue job opportunities with, possibly requiring IT support services.

Implementation

Base metrics include:

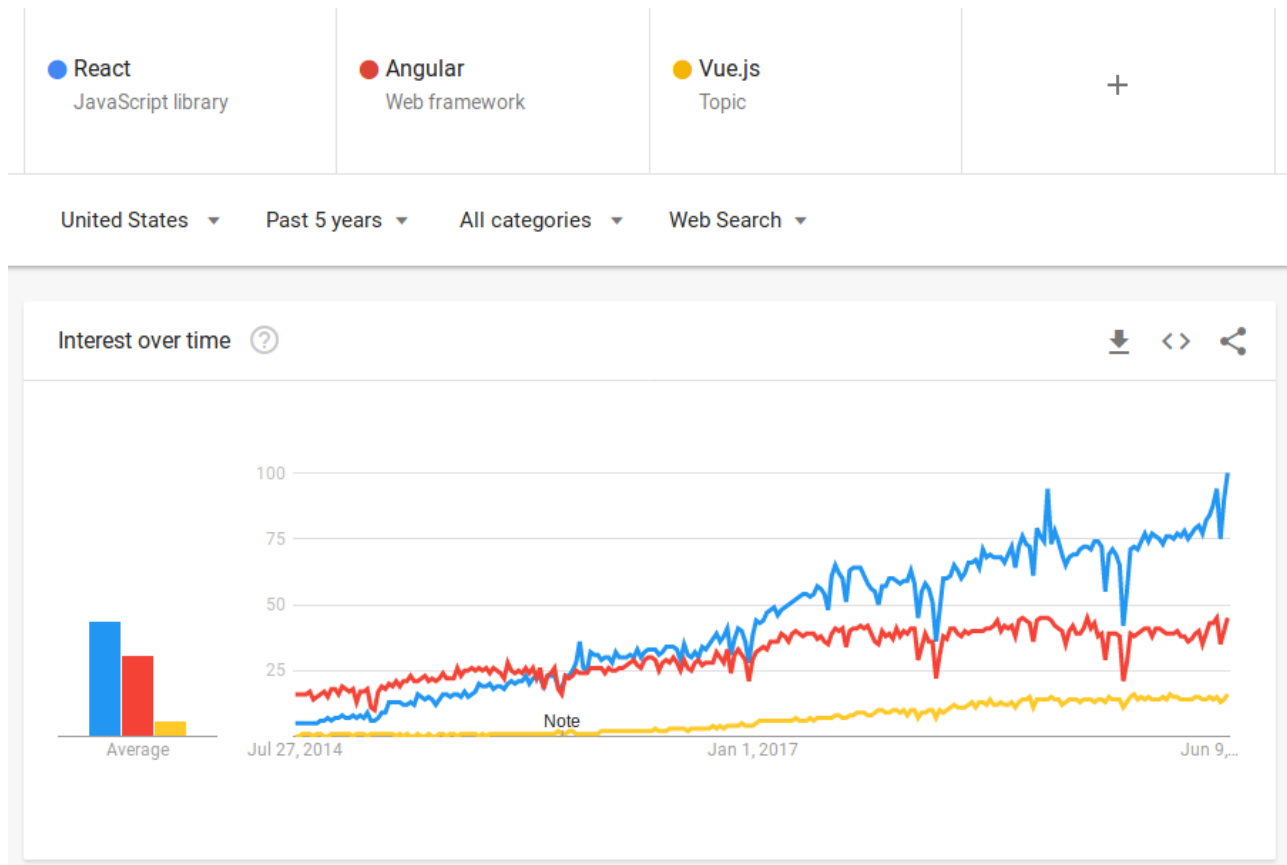
- Number of organizations that created issues for a project
- Number of organizations that created pull requests for a project
- Number of organizations that blog or tweet about a project
- Number of organizations that mention a project in open hiring requests
- Number of organizations that are represented at meetups about this project
- Number of other projects that are dependent on a project
- Number of books about a project
- Google search trends for a project

Visualizations

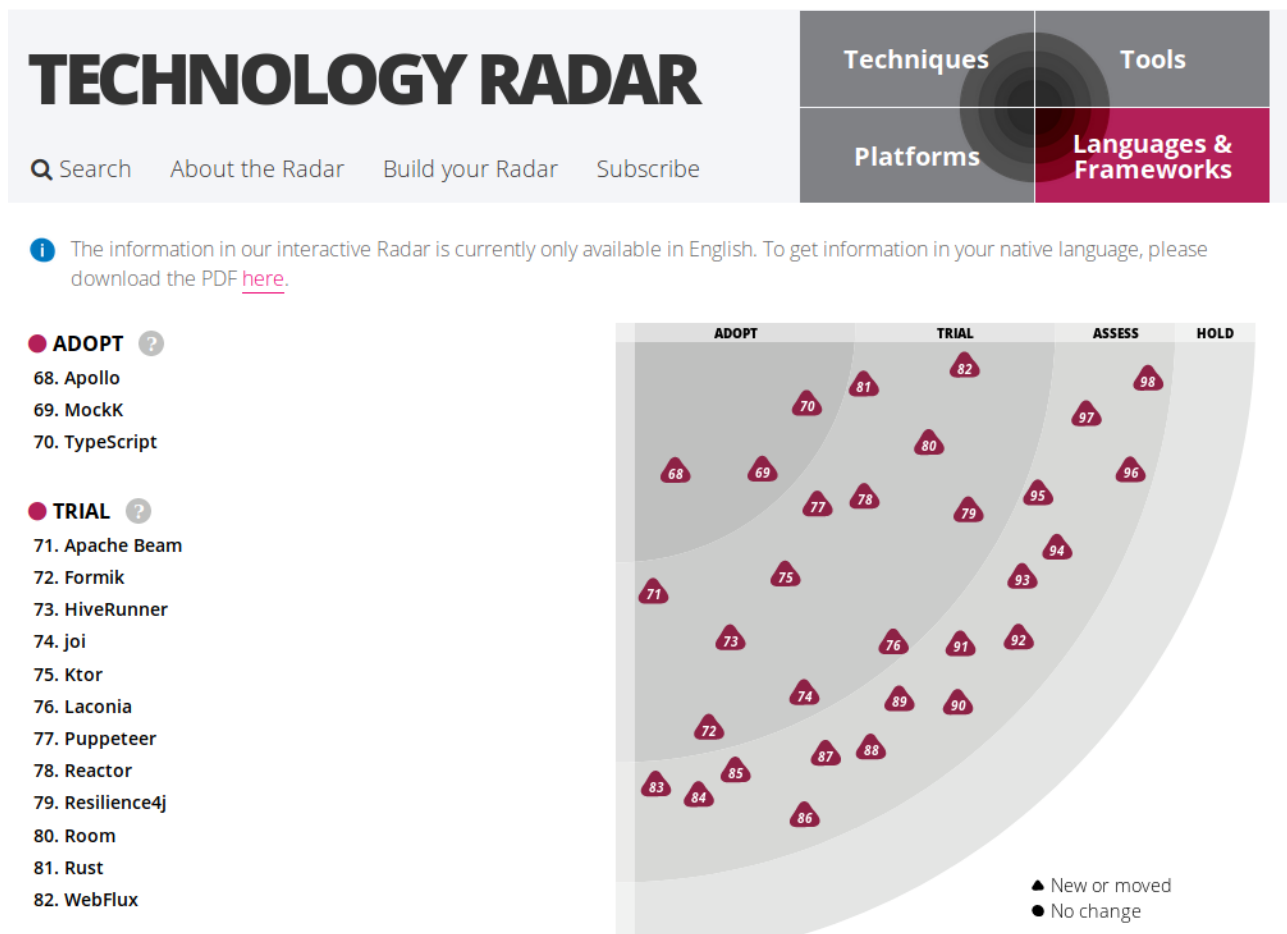
The following visualization demonstrates the number of downstream projects dependent on the project in question. While this visualization does not capture the entirety of the Downstream Organizational Impact of a Project Software metric, it provides a visual for a portion.



Other visualizations could include Google search trends (React vs. Angular vs. Vue.js)



ThoughtWorks publishes a series called 'Tech Radar' that shows the popularity of technologies.



Tech Radar allows you to drill down on projects to see how the assessment has changed over time.

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, composeable 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

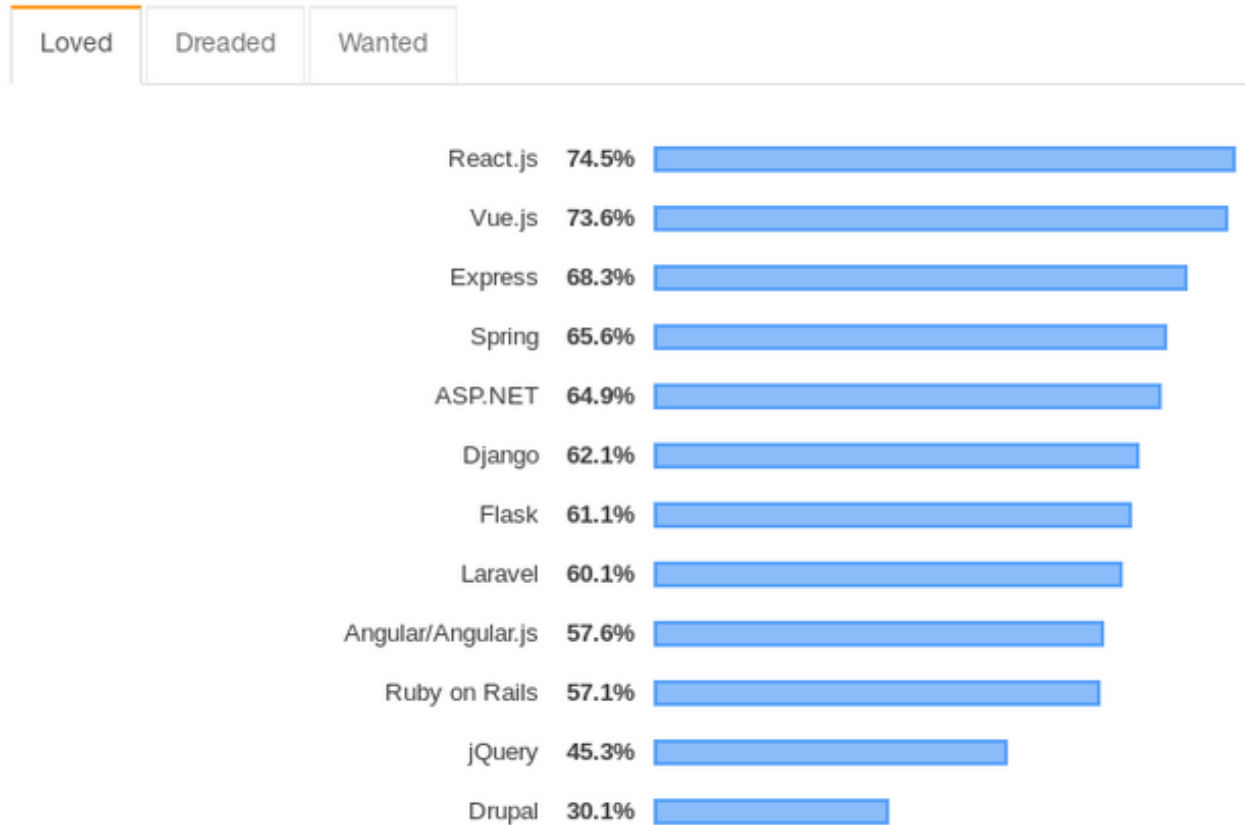
[Understand more »](#)

StackOverview publishes an annual developer's survey



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.

Tools Providing the Metric

- Google Trends - for showing search interest over time
- ThoughtWorks TechRadar - project assessments from a tech consultancy
- StackOverflow Developer's Survey - annual project rankings
- Augur; Examples are available for multiple repositories:
 - [Rails](#)
 - [Zephyr](#)
 - [CloudStack](#)

References

- [Open Source Sponsors](#)
- [Fiscal Sponsors and Open Source](#)
- [Large Corporate OpenSource Sponsors](#)
- [Google Trends API](#)
- [Measuring Open Source Software Impact](#)
- [ThoughtWorks Tech Radar](#)
- [Stack Overflow Developer's Survey](#)

Focus Area - Organizational Value

Goal: Identify if a project is monetarily valuable from an organization's perspective.

Metric	Question
Labor Investment	What was the cost of an organization for its employees to create the counted contributions (e.g., commits, issues, and pull requests)?

Labor Investment

Question: What was the cost of an organization for its employees to create the counted contributions (e.g., commits, issues, and pull requests)?

Description

Open source projects are often supported by organizations through labor investment. This metric tracks the monetary investment of organizations (as evident in labor costs) to individual projects.

Objectives

As organizational engagement with open source projects becomes increasingly important, it is important for organization to clearly understand their labor investment. The objective of this metric is to improve transparency in labor costs for organizations engaged with open source projects. This metric gives an Open Source Program Office (OSPO) manager a way to compare contributed labor costs across a portfolio of projects. For example, the Labor Investment metric can be used to prioritize investment or determine return on investment such as:

- Labor Investment as a means of evaluating OSPO priorities and justifying budgets
- Labor Investment as a way to explain product/program management priority
- Labor Investment as an argument for the value of continued investing in OSPOs
- Labor Investment to report and compare labor costs of contributed vs in-house work
- Labor Investment to compare project effectiveness across a portfolio of projects

Implementation

Base metrics include:

- Number of contributions
- Number of contributions broken out by contributor types (internal / external)
- Number of contributions broken out by contribution types (e.g., commits, issues, pull requests)

Parameters include:

- Hourly labor rate
- Average labor hours to create contribution (by contribution type)

Labor Investment = For each contribution type, sum (Number of contributions * Average labor hours to create contribution * Average hourly rate)

Filters

- Internal vs external contributors
- Issue tags
- Project sources (e.g., internal, open-source repos, competitor open-source repos)

Visualizations

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

Our first visualization of parameterized metrics rely on CSV exports that can be made available from Augur. Spreadsheets are used for metric parameters and calculation formulas. Future implementations may add features for parameter manipulation directly in the webapp.

References

- [Starting an Open Source Program Office](#)
- [Creating an Open Source Program Office](#)
- [Open Source in the Enterprise](#)

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.