

# Open Source Community Health: Analytical Metrics and Their Corresponding Narratives

Sean Goggins  
University of Missouri  
Columbia, Missouri 65211  
outdoors@acm.org

Kevin Lumbard  
University of Nebraska at Omaha  
Omaha, NE 68182  
klumbard@unomaha.edu

Matt Germonprez  
University of Nebraska at Omaha  
Omaha, NE 68182  
germonprez@gmail.com

**Abstract**—Open source projects are most often evaluated by potential contributors and consumers using metrics that describe a level of activity within the project because those measurements are available. The principle question in the minds of most evaluators, however, is "How healthy and sustainable is this project in the context of its competitors or dependent projects"? Limitations of current analysis methods focused on trace data alone are discussed, and reviewed in depth. Next, our methods for conducting engaged field research, developing metrics standards as part of a corporate communal partnership, and molding tools that evolve through a reflexive discourse with practitioners using standard metrics is framed as an approach to consider for examining open source software health and sustainability. Researchers, in particular, need tools for increasing the feasibility of comprehensive, multi-project health and sustainability studies, and connecting trace data with human experience. From a practice perspective, these same conditions are increasing the difficulty organizations and individuals engaged in open source face when trying to understand the status, condition, and health of a particular project, the project's ecosystem or ecosystems emerging around their specific project context. This study examines the work of a Linux Foundation working group, CHAOSS (Community Health Analytics Open Source Software) during the first four years of the formation. The paper concludes with examples of CHAOSS metrics operationalized in partnership with corporate collaborators in a manner that emphasizes comparison, transparency, trajectory and visualization as components for discursive, evolutionary understanding of open source software health.

## I. INTRODUCTION

Open source software is vital to most public computing infrastructure, and an increasingly common component of software development jobs. Following the Heartbleed OpenSSL security bug in 2014, the US Congress sought the aid of the Linux Foundation to help better understand what steps could be taken to make open source project health and risk more visible [1]. Indirectly, open source software's footprint on society is growing in a number of dimensions. As a point of reference, over five billion dollars<sup>1</sup> of direct labor costs are invested in the Linux Foundation's open source ecosystem. Annual revenue estimates for the Linux Kernel alone are currently 50 billion US dollars, supporting a conservative 500 billion US dollar market valuation.<sup>2</sup> Estimates of market value

<sup>1</sup><https://www.linuxfoundation.org/press-release/2015/09/>

<sup>2</sup>IBM acquired Red Hat software, an open source services vendor, for more than 10x revenue in 2018, so we use this as a conservative benchmark of market valuation derived from a multiplier of revenue.

and labor investment are proxies to help situate the present scale of open source software, but financial measures alone are inadequate when we consider how central open source software is within the collected computing infrastructure of the globe.

The demand for coherent, consistent and actionable metrics is growing as a result of corporate engagement with open source [2], [3]. This demand is felt most by open source community managers, who play a role in the formal and informal nurturing of developer communities around open source projects, and internal open source program offices who manage portfolios of open source projects contributed to by a firm [4], [5]. The rapid growth of open source is intersecting with a large number of ongoing metrics development research studies that target individual areas in ecosystems of open source software, but do not yet amount to clearly identifiable, specific metrics, or aid in locating a project's position within particular ecosystems. Everyone involved in open source software wants a clearer understanding of individual projects, and boundaries for ecosystems to advance insight into their joint health and sustainability.

Prior work defines open source project health in a myriad of indicators and tools. Each approach aims to make representations of open source project health efficient through the collection, aggregation and analysis of trace data from repositories, issue trackers, and other traces of work and communication [6]. This study adopts the general view that open source project health is *a project's ability to continue to produce quality software* [7], [8]. Trace data is a building blocks for nearly all measures of open source project health [9], [10], [11], making the collection and analysis of data related to the construction of open source at once essential for representing open source project health, and also at present, incommensurate with the challenge [12], [13]. Specifically, most research to date has three limitations: 1) the research is spread across several disciplines and is therefore not accumulating shared understanding (and arguably is diffusing disconnected insights), 2) most research to date focuses on some specific, narrow segment of the expansive world of open source software, thus forestalling more substantial insights, and 3) analysis of trace artifacts is rarely combined with systematic, ongoing engagement with open source projects as they evolved, limiting the durability of indicators derived from

trace as indicators of health and sustainability.

Though many of the narrowly focused metrics applied to assess open source projects have utility, they do not get at the question of open source project health and sustainability. Metaphorically they could be thought of as using the results of an individual's annual physical to try and assess public health questions for the region they live in. This public health metaphor maps to the use of discrete, open source metrics to what public health researchers refer to as a type III error; we have the right answers to the wrong questions [14]. To avoid type III errors and answer the right questions; open source health and sustainability variation across projects and time, and the presence of specific risk factors must integrate and be weighted against each other based on project context that includes ongoing feedback loops with metric consumers within open source projects. For example, a decline in activity could mean either an emergent risk of the community losing contributor engagement, or the stabilization of a fast-growing new technology. Most cases are not so clear cut as those two.

In this paper, we frame a four-year engaged field research study within the Linux Foundation's Community Health Analytics in Open Source Software (CHAOSS) project. Our findings from this study illustrate a pathway making open source health and sustainability visible in context by emphasizing comparison, transparency, trajectory and visualization through storytelling. Our contribution is to frame an approach for understanding the overall health and sustainability of open source software projects that puts each project in a context salient for its position in its life-cycle, and within groups of projects with similarities identified through a combination of computational similarity measures and heuristic's defined by individuals and organizations engaged in open source software. We also present CHAOSS Augur software as tool to understand community health through storytelling.

## II. LITERATURE REVIEW

### A. Open source project health

Studies of open source health are spread across research communities in software repository mining [15], [16], [10], [17], [18], [19], information systems [11], [2], [20], [21], [22], social computing [23], [24], [25], [26], [12] and software engineering [27], [28]. The aims of open source project health and sustainability metrics (metrics) research differ across disciplines. This opens the second area where current research is incommensurate with the problem of understanding open source health, because most of this research generates narrowly focused metrics on code [10], [29], [30], [31], [32], [33], diversity [34], [35], organizational dynamics [36], [37], [38], [39], risk [40], [41], [18], ecosystem state [17], [42], [43], [38], large scale classification [42], [43], [44], adverse event prediction [45], [18], [19], constructing end user tools [46], [17], [47], [48], and many others.

There is a considerable amount of research constructing and presenting indicators of open source project activity, but a lack of consensus about how indicators derived from trace data might be used to represent a coherent view of open

source project health and sustainability. Researchers define project health through the collection of success measures [49]; including metrics related to project output, process, and the outcomes for project members [11]. In this context, success is the result of project activity and the release of code, otherwise the project is abandoned [50]. Further measures of success explore the growth and diversity of projects [34]. Activity measures tell you when a project is finished; health and sustainability measures would identify that trajectory in advance.

Project health can be framed through measures representing separate but related indications of sustainability and survivability [51]. Sustainability is signaled by three factors – community growth, financial resources, and software management [52]. While sustainability evokes shared commitment, survivability aims to surface indications of a project's vigor, resilience, and organization in the face of risk [53]: when bad things happen to good projects, which ones are most likely to overcome? Sustainability and survivability, together, aspire to provide clear signals of a project's likelihood to continue producing quality software [8].

Likelihood of continued software production is a broad way to frame the goal of assessing open source project health. Measurement of health signals is taken up across a wide range of research domains. Aman et al. [54] applied the Pareto principle to open source projects to define a healthy project as a project where the proportion of core developers to non-core ones equates to roughly 80% of the code contributions being produced by 20% of the active developers. However, only a small slice of open source projects fit with this model [55]. Open source project health has also been assessed by examining the bus factor of projects (e.g., the risk associated with losing key project contributors and the knowledge that they possess) [56], [57], and research has shown that many open source projects have low bus factors, implying that they are not likely to survive disruption. In the next section we review a wide range of prior open source software research adjacent to questions of health, drawing out our argument that the present state of research is incommensurate with the problem space.

### B. Open Source Health Research Solves a Different Problem

Researchers explore open source project health in different contexts and employ many different measures. Additionally, researchers often apply measures from other domains that don't take into account the unique nature of open source projects. What is clear is that activity is a common proxy for understanding project health [8], however, research focused on discrete activities addresses only the presence of activity, offering little insight about the complexities of future sustainability or survivability. Activity metrics illustrate the likelihood that a project will ever get traction, but tells us little about its arc following launch.

Activity is insufficient as a proxy for open source health particularly because many open source health studies are conducted at the smallest unit of analysis, building metrics up from one type of repository activity, such as a software

commit [58]. Alternately, large scale summaries of open source health often squash entire repository histories and draw inferences about projects and ecosystems [17] without consideration of the evolution of projects over time. Table I illustrates the wide variety of measures used to operationalize open source project health using activity metrics or large scale summaries. Much of the early literature on open source project health focused on success measures [59], [60], [11], [61], [62], [63], however as open source health research has matured, the focus has moved to measures of sustainability [64], [51], [65], [52], [66], [67], [68], [58] and further, to an understanding that open source health must include considerations for social interactions and project diversity when estimating survivability [69], [7], [70], [34], [9], [71], [72]. Sustainability and survivability are difficult to estimate based on the current or historical level of activity alone, especially in an era where corporate engagement is a rapidly increasing aspect of open source ecosystems across the globe [2].

Our analysis of the literature surfaces a number of discrete metrics aimed at different aspects of open source project health, including a) sufficient scale, b) project culture, c) process quality, d) product quality, e) contextualized risk, f) license risk, and g) corporatization and access to resources. Each set of discrete metrics is useful, and reflects a particular aspect of an open source project that can contribute to its likelihood of continuing to produce quality software. This contemporary focus on the development of discrete metrics from open source project activity is necessary for ascertaining project health but does not make project health visible. While open source practitioners yearn for insight, they are often confused by an overwhelming array of dashboards that force together some set of "frankenmetrics" derived from the categories we identify. Open source metrics research is discrete, conducted at a point in time and not focused on the specific needs of metrics consumer communities like open source community managers and open source program offices. The "measurement of open source project health is further confounded because health metrics may have different meanings for different projects and there is little consensus about the correct way to calculate metrics across projects" [2, p. 14].

### III. METHODS

Our work is built upon an ongoing ten-year research study exploring organizational engagement with open source projects. Our efforts are localized in open source projects that include heavy organizational engagement, mainly projects brokered by the Linux Foundation,<sup>3</sup> a 501(c)(6) trade association and its member companies, which constitute a substantial majority of the technology sector of the economy. The Linux Foundation has helped "establish, build, and sustain some of the most critical open source technologies fostering innovation in every layer of the software stack. The Linux Foundation hosts projects spanning enterprise IT, embedded systems, consumer electronics, cloud, and networking".

<sup>3</sup><https://www.linuxfoundation.org/projects/linux/>

The methods we use systematically integrate field engagement with analysis of trace data from open source project activity to build greater understanding of the relationship between how projects and ecosystems experience health, and what analysis of open source software trace data can reveal. Within this context and over the prior ten years, we employed a variety of approaches including participant observation [77], group informatics [6] (a systematic methodological approach for reflexive analysis of field data and trace data), direct engagement [78], and critical reflection [79]. Participant observation is used as a field-based approach when members of our research team were directly engaged in the practices we sought to understand [77], [80], [78]. Group informatics was used to reflexively make meaning from the intersection of our fieldwork and the significant amounts of digital trace data that emerges within open source projects; thus ensuring coherence of research constructs in the outcomes produced in our participant observation [23], [6]. Finally, to ground our findings from participant observation [77] and group informatics [6], we used direct engagement and critical reflection as our "process of learning from experience" [79]. Through this field study, data was generated from approximately 200 interviews, 200 survey responses, ten focus groups, 1,000 pages of field notes, constant comparison, content analysis [81], trace ethnography [13], social network analysis [61], [6], and computational linguistic analysis [82].

#### A. Engaged Field Research

This paper focuses on a four-year engagement with the CHAOSS project<sup>4</sup>. Since its inception at the Linux Foundation's Open Source Leadership Summit in Spring 2017, CHAOSS has come to include members from 70+ organizations such as RedHat, VMWare, Intel, Mozilla, The Eclipse Foundation, Oath, and Bitergia. The mission of the CHAOSS project is to:

- 1) Produce integrated, open source software for analyzing software development, and definition of standards and models used in that software in specific use cases
- 2) Establish implementation-agnostic open source project health indicators for measuring community activity, contributions, and health.
- 3) Optionally, produce standardized open source project health indicators exchange formats, reports, badging, use cases, models, or recommendations to analyze specific open source issues.

### IV. ORDER FROM CHAOSS

The CHAOSS communities efforts to develop a set of metrics and the tooling around metrics is complicated by competing perspectives. The standards the community can develop are limited to the discrete metrics that can be discussed and agreed to, which is not of course the end of the line for defining health and sustainability in open source.

<sup>4</sup><http://chaoss.community>

Term Used	Operationalization of Project Health	Published
Success	(a) Number of contributors	2002 [59]
Success	(b) Popularity on the web	2005 [60]
Success	(c) Measures of the Output of Systems Development, Measures of the Process of Systems Development, and Effects on Project Teams.	2006 [11]
Success	(c) Codification of the messages recorded in the bug tracking system	2008 [61]
Success	(c) Software quality, community service quality, and user satisfaction	2009 [62]
Success	(f) Open Source License selection	2009 [63]
Success	(c) Quality model - Size and regeneration adequacy, Interactivity and workload adequacy, and Composition adequacy	2010 [73]
Success	(a) Project popularity and developer activity	2012 [74]
Success	(a) Long-term involvement of users and developers, project continually develops, matures, and evolves over time	2012 [75]
Success	(a) Number of users and developers	2012 [76]
Success	(a) Long-term involvement of users and developers, project continually develops, matures, and evolves over time	2012 [75]
Success	(a) Number of users and developers	2012 [76]
Success	(a) Long-term involvement of users and developers, project continually develops, matures, and evolves over time	2012 [75]
Success	(a) Number of users and developers	2012 [76]
Success	(b) Diversity - separation, variety, and disparity	2013 [69]
Success	(a) Pareto principle - 80% of contributions come from 20% of contributors	2015 [55]
Success	(b) Project growth and social diversity	2016 [34]
Success	(d) Low software coupling and high interactive discussion	2016 [9]
Success	(b) Concentration Index - Openness	2016 [71]
Sustainability	(d) Quality model - product quality, process maturity and sustainability	2009 [64]
Sustainability	(a) Development base (size), project age and the size of niche	2010 [51]
Sustainability	(c) Patch contribution and patch process	2010 [65]
Sustainability	(g) Community growth, financial resources and software management	2011 [52]
Survivability	(b) Viability index - vigor, resilience, and organization	2012 [53]
Sustainability	(b) Relationships among people, technologies, and organizations.	2012 [66]
Sustainability & Success	(b) Virtuous Circle - Feedback loop of open source best practices	2013 [67]
Sustainability	(e) Ability to Fork	2013 [68]
Sustainability	(a) Commits, retention of committers	2014 [58]
Risk	Truck factor - Number of key people that need to leave for a project to stall due to lack of knowledge	2011 [57]
Risk	(a) Bus Factor - Number of key people that need to leave for a project to stall due to lack of knowledge	2015 [56]
Health	(e) Productivity, robustness, and Niche creation	2014 [17]
Health	(b) Positive experience, trust in the leadership of the project leader, the demonstration of reciprocity, marketing the community, enriching knowledge, and face-to-face meetings	2015 [7]
Health	(b) Social Health - Ability to get help	2016 [70]
Health	(a) Pareto principle - 80% of contributions come from 20% of contributors	2017 [54]
Health	(b) Gender Bias - Ratio of female to male contributions accepted	2017 [72]

TABLE I

OPEN SOURCE HEALTH IS EXPLORED IN WIDE VARIETY OF WAYS. (A) SUFFICIENT SCALE, (B) PROJECT CULTURE, (C) PROCESS QUALITY, (D) PRODUCT QUALITY, (E) CONTEXTUALIZED RISK, (F) LICENSE RISK, (G) CORPORATIZATION AND ACCESS TO RESOURCES.

1) *Project Stakeholders*: Work within the CHAOSS project identified two key stakeholder types - community managers and program managers - interested in community health metrics. Open source community managers oversee communities in the communal side of the corporate communal relationship. The job of the open source community manager is to enable open source project members to achieve personal and project goals [83]. The role does not always have an explicit title nor is the job mutually exclusive. These individuals may take on many roles in a project, including developers, document writers, facilitators and maintainers. The focus for these individuals is building and maintain healthy communities of contributors - towards the successful completion of individual project goals.

Open source program managers oversee corporate interests from corporate side of corporate communal relationships. The key difference between an open source program manager and a community manager is the corporate focus. Open source program managers are primarily concerned with the interests of the corporation and may manage a portfolio of open source projects that are strategically important to the corporation. The

responsibilities of an open source program manager are varied but often include - communicating the open source strategy within and outside the company, overseeing the execution of the corporate strategy, facilitating the effective use of open source in within the company, ensuring the quality and frequent release of software, engaging with communities and ensuring that the company contributes to open source projects, license compliance and fostering an open source culture [84].

Community and program managers take a variety of perspectives, depending on where their communities are in the life-cycle of growth, maturity, and decline. This paper is an evolving report of what we are learning from community and program office managers, some of whom we are working with on live experiments with a CHAOSS project prototyping software tool called Augur<sup>5</sup>. At this point, we are paying particular attention to how community managers consume metrics and how the presentation of open source project health and sustainability metrics could make those metrics more and in some cases less useful for evaluating open source project

<sup>5</sup><http://www.github.com/CHAOSS/augur>

health.

2) *New Goals and Concerns*: "Who is going to use these metrics and for what purpose?" is a question that matters a great deal to the people engaged in the CHAOSS project. The development of health and sustainability metrics for open source software surfaced specific concerns among our participants. Making statistics available to people unfamiliar with the state, functioning and current goals of an open source ecosystem creates risk that the numbers will lead to a reflexive, uncontextualized side narrative that could harm the efforts of open source community and project office managers. The risk is that numbers without context will aid in the development of stories that are not focused on increasing health and sustainability, but instead focused on telling stories that emphasize activity over health. Through discussions with community managers and program managers we have identified several user stories related to project health:

- "As a community manager I want to be able to compare the open source projects that I manage"
- "As a community manager I want to identify trends in the data collected about my repositories"
- "As a community manager I need to know which contributors are making the largest impact across projects that I manage"
- "As a community manager I need to know how contributor behavior is changing"
- "As a program manager adopting open source, I need to know what risks my company will face if they use software"
- "As a program manager adopting open source, I need to be able to compare projects to determine which solution will be the least risky to our company"

3) *From Metrics to Health*: Frustrations with the elusiveness of coherent, durable indicators of open source project health pervade the experiences we observed. In practice, there is evidence that community managers and open source program offices occasionally stand up discrete metrics tools, but ultimately fail to find them useful in their search to fully understand open source project health over time. One of the central goals that emerged from our participatory design process was to make it easier for open source stakeholders to "get their bearings" on a project and understand "how things are going". Interactions with stakeholders created insight that this was most easily accomplished when comparisons between internal and external projects over time, are readily available.

Through our field study, four core principles for making sense of data emerged: comparison, transparency, trajectory and visualization. These principles can be operationalized through four human-centered data science strategies:

- 1) *Comparison*: Enabling comparisons, letting people navigate complex unknowns analogically as well as see how their project compares with others they are familiar with.
- 2) *Trajectory*: Making time a fundamental dimension in all open source project health indicators as point in time scores are useful.

3) *Transparency*: Making the provenance from raw data to open source project health indicators visualization transparent. People trust open source project health indicators when they can see the underlying data and providing traceability back to the CHAOSS project open source project health indicators requires transparency.

4) *Visualization*: Enabling visualizations as downloadable as a .csv, .svg, or other data exchange format from which to build local stories.

Great answers to the wrong questions are more commonplace than we prefer because open source software work is evolving quickly and we do not yet have a list of the right questions for many specific project situations. To ground how design strategies are employed in our study with CHAOSS we frame our iterative inquiries around a set of specific, common design tools and contextual questions:

- 1) *Goals*: What are metrics going to help you accomplish?
- 2) *Use Cases*: When you go to use metrics, what are the use cases you have? A case can be simple, ill-formed and even 'unpretty':
  - a) "My manager wants to know if anyone else is working on this project?"
  - b) "It seems like my community is leveling off? Is it? Or is it just so large now I cannot tell?"

Fundamentally, the community managers and program offices we talked with are clear that they ultimately want to be able to use tools (like Augur) to construct narrative descriptions of the nuanced status for their projects. The story of open source project health is at its core a story, not a dashboard of discrete metrics. And the story cannot be told without discrete metrics, available in a form that follows the design principles that emerged from our engaged field work.

## V. USING AUGUR TO TELL STORIES

Our engaged field work led to several dozen implementations of Augur in the service of large corporations in the technology sector over the past three years. Through each engagement, the data collected, CHAOSS metrics generated, and integration of those metrics into visualizations evolved. Organizations like Twitter simply leveraged Augur's verifiable data completeness to build their own front-ends. We view the tool, and its evolution, as both a means to an ends, and a reflection of how the CHAOSS community's understanding of the potential of standard metrics is evolving.

Augur empowers open source community managers to tell data-driven stories. Augur is designed to enable the rapid, reflexive exploration project health using CHAOSS metrics to answer questions as they emerge from experience, and used to understand specific questions about health and sustainability within an organizational context, including open source project ecosystems and corporations. This is possible due to its well documented, relational data model, and software that verifies data collection accuracy<sup>6</sup>.

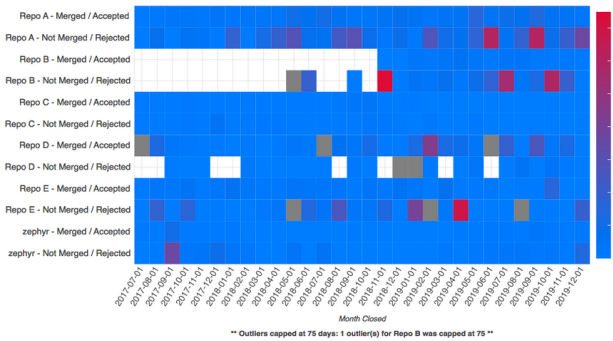
<sup>6</sup><https://www.augurlabs.io/wp-content/uploads/2020/10/20201025-Release-0.14.0.svg>

Telling a health and sustainability story for an open source project requires more than a collection of the metrics available and defined for a project. Because having tools that are useful for telling stories is of such importance to open source program and community managers the visual design of data in Augur is aimed at providing a synthesis of several related metrics in visualizations that solve specific problems or answer specific questions that emerged during design. Two examples of Augur’s use of metrics to synthesize narrative understanding of open source projects through analysis of trace data make the approach more clear by synthesizing collections of CHAOSS metrics into captioned visualizations for a half dozen different open source projects: 1) Understanding new contributors and repeat contributors over time, and 2) Understanding pull request responsiveness within a project, contrasted with responsiveness from other projects.

### A. Example: Pull Request Responsiveness Comparisons

Responsiveness to outside contributions, reflected as pull requests on GitHub, are one way the both community managers and open source program officers draw comparisons between different projects they are contributing to or competing against. The use of visual contrast in this figure helps illustrate differences between one project, Zephyr, and an anonymized field of competitors. Augur’s anonymization capability in comparisons enables contextualized understanding without directly commenting on other projects.

Fig. 1. An example of anonymized comparisons between projects in either a competitive or complementary ecosystem relationship.

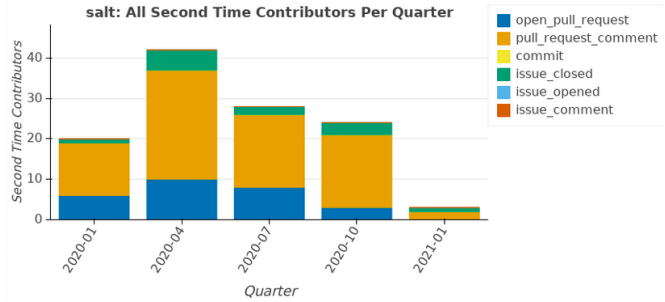


### B. Example: Understanding New and Repeat Contributors

Another significant, and complex indicator of project health included a number of CHAOSS metrics related to new contributors, including the types of contributions, and understanding of when or if second contributions were made. Augur provides individual, or comparative analysis of these new and repeat contributor statistics, using parameterizable, visualization API calls. In figure 2, we see a time series graph of the types of *second* contributions made on a project.

Augur’s design emerged from a stark recognition that a good deal of open source metrics research to date has not arrived at the central importance of combining trace analysis with storytelling, and systematic adaptation of metric combinations as an evolving enterprise.

Fig. 2. An example analysis of the types of contributions provided by second time contributors during quarterly periods for the salstack/salt project.



## VI. DISCUSSION

As researchers, we build tools that help us gather data, and report findings in academic venues. In our role as engaged field researchers, we made a decision to ensure that our collection of trace data, and its integration with field work, is not the end goal. Instead, in this case, collaborators within the CHAOSS project are guiding the evolution of the tool. First and foremost, its data gathering, and relational organization, enable new questions to be answered with relative ease. Augur data can be accessed and presented through Jupyter notebooks, two different web front-ends of, and customized front-ends that access Augur’s RESTFUL API. The API endpoints follow a logic that is twofold. First, each CHAOSS metric is an individual endpoint. Second, creating new endpoints that generate data, or static visualization files makes it possible to ask nearly any question you can think of, across thousands of open source software repositories, in hours to days.

Every computing centered open source software scholar is capable of accomplishing similar working systems. However, there are few to date, and no others whose origins coexist with those of a coordinated effort to develop reusable open source software metrics around a range of specific concerns. And, as scholars, we too often leave the tools we build by the wayside as funding shifts, and our aims move on. Our position, and our hope, is that by sharing our approach, aims, and accomplishments to date, this work will inspire a discussion about how to make research software aimed at providing a coherent view of open source software health highly adaptable to changes, and sustainable.

### A. Limitations

This paper shares a sliver of the technical depth and breadth of an open source software project aimed at making CHAOSS metrics actionable. The argument we make, is that an approach focused on engaged field research, and sustained discussions with practitioners, requires sustained commitment to the development of tools like Augur. Centering the aims of Augur on discourse about open source project health, and narrative construction for CHAOSS organizations, limits discrete empirical findings in this paper. We aim to report those findings, in partnership with our CHAOSS collaborators, in future work.

## REFERENCES

- [1] G. Walden and G. Harper, "Letter to Mr. Zemlin from the One Hundred Fifteenth Congress of the United States Committee on Energy and Commerce," Apr. 2018. [Online]. Available: <https://energycommerce.house.gov/wp-content/uploads/2018/04/040218-Linux-Evaluation-of-OSS-Ecosystem.pdf>
- [2] M. Germonprez, G. J. Link, K. Lumbard, and S. Goggins, "Eight Observations and 24 Research Questions About Open Source Projects: Illuminating New Realities," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 57:1–57:22, Nov. 2018. [Online]. Available: <http://doi.acm.org/10.1145/3274326>
- [3] M. Germonprez, J. E. Kendall, K. E. Kendall, L. Mathiassen, B. Young, and B. Warner, "A theory of responsive design: A field study of corporate engagement with open source communities," *Information Systems Research*, vol. 28, no. 1, pp. 64–83, 2017.
- [4] A. Foundation, "Apache Community Development," 2019. [Online]. Available: <https://community.apache.org/>
- [5] M. S. Elliott and W. Scacchi, "Free software developers as an occupational community: resolving conflicts and fostering collaboration," in *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*. ACM, 2003, pp. 21–30.
- [6] S. P. Goggins, C. Mascaro, and G. Valetto, "Group informatics: A methodological approach and ontology for sociotechnical group research," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 3, pp. 516–539, 2013.
- [7] D. Naparat, P. Finnegan, and M. Cahalane, "Healthy community and healthy commons: 'Opensourcing' as a sustainable model of software production," *Australasian Journal of Information Systems*, vol. 19, no. 0, 2015, bibtext: Naparat2014.
- [8] G. J. Link and M. Germonprez, "Assessing Open Source Project Health," *AMCIS 2018 Proceedings*, Aug. 2018.
- [9] S. Daniel and K. Stewart, "Open source project success: Resource access, flow, and integration," *The Journal of Strategic Information Systems*, vol. 25, no. 3, pp. 159–176, Oct. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0963868716000093>
- [10] J. Davies, D. M. German, M. W. Godfrey, and A. Hindle, "Software Bertillonage: Finding the Provenance of an Entity," in *Proceedings of the 8th Working Conference on Mining Software Repositories*, ser. MSR '11. New York, NY, USA: ACM, 2011, pp. 183–192. [Online]. Available: <http://doi.acm.org/10.1145/1985441.1985468>
- [11] K. Crowston, J. Howison, and H. Annabi, "Information systems success in free and open source software development: Theory and measures," *Software Process: Improvement and Practice*, vol. 11, no. 2, pp. 123–148, 2006.
- [12] N. McDonald, K. Blincoe, E. V. A. Petakovic, and S. Goggins, "Modeling distributed collaboration on Github," *Advances in Complex Systems*, vol. 17, no. 07n08, p. 1450024, 2014.
- [13] R. S. Geiger and D. Ribes, "Trace ethnography: Following coordination through documentary practices," in *2011 44th Hawaii International Conference on System Sciences*. IEEE, 2011, pp. 1–10.
- [14] S. Schwartz and K. M. Carpenter, "The right answer for the wrong question: consequences of type III error for public health research," *American Journal of Public Health*, vol. 89, no. 8, pp. 1175–1180, Aug. 1999. [Online]. Available: <http://ajph.aphapublications.org/doi/10.2105/AJPH.89.8.1175>
- [15] R. D. Arnold and J. P. Wade, "A definition of systems thinking: a systems approach," *Procedia Computer Science*, vol. 44, pp. 669–678, 2015.
- [16] A. Begel, J. D. Herbsleb, and M.-A. Storey, "The future of collaborative software development," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work Companion*. ACM, 2012, pp. 17–18.
- [17] S. Jansen, "Measuring the health of open source software ecosystems: Beyond the scope of project health," *Information and Software Technology*, vol. 56, no. 11, pp. 1508–1519, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950584914000871>
- [18] L. Madeyski and M. Kawalerowicz, "Continuous Defect Prediction: The Idea and a Related Dataset," in *Proceedings of the 14th International Conference on Mining Software Repositories*, ser. MSR '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 515–518. [Online]. Available: <https://doi.org/10.1109/MSR.2017.46>
- [19] T. Rausch, W. Hummer, P. Leitner, and S. Schulte, "An Empirical Analysis of Build Failures in the Continuous Integration Workflows of Java-based Open-source Software," in *Proceedings of the 14th International Conference on Mining Software Repositories*, ser. MSR '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 345–355. [Online]. Available: <https://doi.org/10.1109/MSR.2017.54>
- [20] D. Nakikj and L. Mamykina, "Lost in migration: Information management and community building in an online health community," ACM Press, 2018, pp. 1–14. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3173574.3173720>
- [21] J. D. Wells, J. S. Valacich, and T. J. Hess, "What signal are you sending? How website quality influences perceptions of product quality and purchase intentions," *MIS quarterly*, pp. 373–396, 2011.
- [22] A. Dimoka, Y. Hong, and P. A. Pavlou, "On product uncertainty in online markets: Theory and evidence," *MIS quarterly*, vol. 36, 2012.
- [23] K. Blincoe, J. Sheoran, S. Goggins, E. Petakovic, and D. Damian, "Understanding the popular users: Following, affiliation influence and leadership on GitHub," *Information and Software Technology*, vol. 70, pp. 30–39, 2016.
- [24] M. Cataldo, M. Bass, J. D. Herbsleb, and L. Bass, "On coordination mechanisms in global software development," in *International Conference on Global Software Engineering (ICGSE 2007)*. IEEE, 2007, pp. 71–80.
- [25] P. Wagstrom, J. D. Herbsleb, R. E. Kraut, and A. Mockus, "The impact of commercial organizations on volunteer participation in an online community," in *Academy of Management Annual Meeting*, 2010.
- [26] M. Cataldo and J. D. Herbsleb, "Communication networks in geographically distributed software development," ACM Press, 2008, p. 579. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1460563.1460654>
- [27] D. M. German, Y. Manabe, and K. Inoue, "A sentence-matching method for automatic license identification of source code files," in *Proceedings of the IEEE/ACM international conference on Automated software engineering*. ACM, 2010, pp. 437–446.
- [28] J. M. Gonzalez-Barahona, P. Sherwood, G. Robles, and D. Izquierdo, "Technical lag in software compilations: Measuring how outdated a software deployment is," in *IFIP International Conference on Open Source Systems*. Springer, Cham, 2017, pp. 182–192.
- [29] D. Izquierdo-Cortazar, N. Sekitoleko, J. M. Gonzalez-Barahona, and L. Kurth, "Using Metrics to Track Code Review Performance," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE'17. New York, NY, USA: ACM, 2017, pp. 214–223. [Online]. Available: <http://doi.acm.org/10.1145/3084226.3084247>
- [30] K. Johari and A. Kaur, "Validation of Object Oriented Metrics Using Open Source Software System: An Empirical Study," *SIGSOFT Softw. Eng. Notes*, vol. 37, no. 1, pp. 1–4, Jan. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2088883.2088893>
- [31] M. Mondal, C. K. Roy, and K. A. Schneider, "Connectivity of Co-changed Method Groups: A Case Study on Open Source Systems," in *Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research*, ser. CASCON '12. Riverton, NJ, USA: IBM Corp., 2012, pp. 205–219. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2399776.2399795>
- [32] F. Rahman and P. Devanbu, "How, and Why, Process Metrics Are Better," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 432–441. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2486788.2486846>
- [33] M. Sharma, M. Kumari, and V. B. Singh, "Post release versions based code change quality Metrics," in *Proceedings of the Third International Symposium on Women in Computing and Informatics*, ser. WCI '15. New York, NY, USA: ACM, 2015, pp. 235–243. [Online]. Available: <http://doi.acm.org/10.1145/2791405.2791466>
- [34] J. Aué, M. Haisma, K. F. Tómasdóttir, and A. Bacchelli, "Social Diversity and Growth Levels of Open Source Software Projects on GitHub," in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '16. New York, NY, USA: ACM, 2016, pp. 41:1–41:6. [Online]. Available: <http://doi.acm.org/10.1145/2961111.2962633>
- [35] N. Eghbal, "Roads and bridges: The unseen labor behind our digital infrastructure," Ford Foundation, Tech. Rep., 2016.
- [36] A. Capiluppi, A. Serebrenik, and A. Youssef, "Developing an H-index for OSS Developers," in *Proceedings of the 9th IEEE Working Conference on Mining Software Repositories*, ser. MSR '12. Piscataway, NJ, USA: IEEE Press, 2012, pp. 251–254. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2664446.2664484>

- [37] A. Meneely, A. C. R. Tejada, B. Spates, S. Trudeau, D. Neuberger, K. Whitlock, C. Ketant, and K. Davis, "An Empirical Investigation of Socio-technical Code Review Metrics and Security Vulnerabilities," in *Proceedings of the 6th International Workshop on Social Software Engineering*, ser. SSE 2014. New York, NY, USA: ACM, 2014, pp. 37–44. [Online]. Available: <http://doi.acm.org/10.1145/2661685.2661687>
- [38] K. Yamashita, S. McIntosh, Y. Kamei, and N. Ubayashi, "Magnet or Sticky? An OSS Project-by-project Typology," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: ACM, 2014, pp. 344–347. [Online]. Available: <http://doi.acm.org/10.1145/2597073.2597116>
- [39] A. Youssef and A. Capiluppi, "The Impact of Developer Team Sizes on the Structural Attributes of Software," in *Proceedings of the 14th International Workshop on Principles of Software Evolution*, ser. IWVSE 2015. New York, NY, USA: ACM, 2015, pp. 38–45. [Online]. Available: <http://doi.acm.org/10.1145/2804360.2804365>
- [40] K. Johari and A. Kaur, "Effect of Software Evolution on Software Metrics: An Open Source Case Study," *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 5, pp. 1–8, Sep. 2011. [Online]. Available: <http://doi.acm.org/10.1145/2020976.2020987>
- [41] I. Kaur and A. Kaur, "Empirical Study of Software Quality Estimation," in *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, ser. CCSEIT '12. New York, NY, USA: ACM, 2012, pp. 694–700. [Online]. Available: <http://doi.acm.org/10.1145/2393216.2393332>
- [42] M. Ortu, G. Destefanis, M. Kassab, and M. Marchesi, "Measuring and Understanding the Effectiveness of JIRA Developers Communities," in *Proceedings of the Sixth International Workshop on Emerging Trends in Software Metrics*, ser. WETSoM '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 3–10. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2821491.2821495>
- [43] K. Plakidas, S. Stevanetic, D. Schall, T. B. Ionescu, and U. Zdun, "How Do Software Ecosystems Evolve? A Quantitative Assessment of the R Ecosystem," in *Proceedings of the 20th International Systems and Software Product Line Conference*, ser. SPLC '16. New York, NY, USA: ACM, 2016, pp. 89–98. [Online]. Available: <http://doi.acm.org/10.1145/2934466.2934488>
- [44] F. Taibi, "Reusability of Open-source Program Code: A Conceptual Model and Empirical Investigation," *SIGSOFT Softw. Eng. Notes*, vol. 38, no. 4, pp. 1–5, Jul. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2492248.2492276>
- [45] I. Abunadi and M. Alenezi, "Towards Cross Project Vulnerability Prediction in Open Source Web Applications," in *Proceedings of the The International Conference on Engineering & MIS 2015*, ser. ICEMIS '15. New York, NY, USA: ACM, 2015, pp. 42:1–42:5. [Online]. Available: <http://doi.acm.org/10.1145/2832987.2833051>
- [46] A. Hannemann, K. Liiva, and R. Klamma, "Navigation Support in Evolving Open-Source Communities by a Web-Based Dashboard," in *Open Source Software: Mobile Open Source Technologies*, ser. IFIP Advances in Information and Communication Technology, L. Corral, A. Sillitti, G. Succi, J. Vlasenko, and A. I. Wasserman, Eds. Springer Berlin Heidelberg, May 2014, no. 427, pp. 11–20. [Online]. Available: [http://link.springer.com/chapter/10.1007/978-3-642-55128-4\\_2](http://link.springer.com/chapter/10.1007/978-3-642-55128-4_2)
- [47] C. Rosen, B. Grawi, and E. Shihab, "Commit Guru: Analytics and Risk Prediction of Software Commits," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2015. New York, NY, USA: ACM, 2015, pp. 966–969. [Online]. Available: <http://doi.acm.org/10.1145/2786805.2803183>
- [48] D. Rozenberg, I. Beschastnikh, F. Kosmale, V. Poser, H. Becker, M. Palyart, and G. C. Murphy, "Comparing Repositories Visually with Repograms," in *Proceedings of the 13th International Conference on Mining Software Repositories*, ser. MSR '16. New York, NY, USA: ACM, 2016, pp. 109–120. [Online]. Available: <http://doi.acm.org/10.1145/2901739.2901768>
- [49] K. Crowston, H. Annabi, and J. Howison, "Defining open source software project success," in *Proceedings of the 24th International Conference on Information Systems ICIS 2003*, 2003, pp. 327–340.
- [50] C. M. Schweik and R. C. English, *Internet success: a study of open-source software commons*. Cambridge, Mass.: MIT Press, 2012.
- [51] I. Chengalur-Smith, A. Sidorova, and S. Daniel, "Sustainability of Free/Libre open source projects: A longitudinal study," *Journal of the Association for Information Systems*, vol. 11, no. 11, Nov. 2010.
- [52] F. L. Arantes and F. M. P. Freire, "Aspects of an open source software sustainable life cycle," in *IFIP International Conference on Open Source Systems*. Springer, 2011, pp. 325–329.
- [53] U. Raja and M. J. Tretter, "Defining and evaluating a measure of open source project survivability," *IEEE Transactions on Software Engineering*, vol. 38, no. 1, pp. 163–174, Jan. 2012.
- [54] H. Aman, A. E. Burhandenny, S. Amasaki, T. Yokogawa, and M. Kawahara, "A Health Index of Open Source Projects Focusing on Pareto Distribution of Developer's Contribution," in *2017 8th International Workshop on Empirical Software Engineering in Practice (IWSEPE)*, Mar. 2017, pp. 29–34.
- [55] K. Yamashita, S. McIntosh, Y. Kamei, A. E. Hassan, and N. Ubayashi, "Revisiting the applicability of the pareto principle to core development teams in open source software projects," in *Proceedings of the 14th International Workshop on Principles of Software Evolution*. ACM, 2015, pp. 46–55.
- [56] V. Cosentino, J. L. C. Izquierdo, and J. Cabot, "Assessing the bus factor of git repositories," in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Mar. 2015, pp. 499–503.
- [57] M. Torchiano, F. Ricca, and A. Marchetto, "Is My Project's Truck Factor Low?: Theoretical and Empirical Considerations About the Truck Factor Threshold," in *Proceedings of the 2Nd International Workshop on Emerging Trends in Software Metrics*, ser. WETSoM '11. New York, NY, USA: ACM, 2011, pp. 12–18. [Online]. Available: <http://doi.acm.org/10.1145/1985374.1985379>
- [58] J. Gamalielsson and B. Lundell, "Sustainability of open source software communities beyond a fork: How and why has the libreoffice project evolved?" *Journal of Systems and Software*, vol. 89, pp. 128–145, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121213002744>
- [59] S. Krishnamurthy, "Cave or Community? An Empirical Examination of 100 Mature Open Source Projects," *First Monday*, vol. 7, no. 6, Jun. 2002. [Online]. Available: <http://firstmonday.org/ojs/index.php/fm/article/view/960>
- [60] D. Weiss, "Measuring success of open source projects using web search engines," Genova, Italy, 2005.
- [61] K. Crowston and B. Scozzi, "Bug fixing practices within Free/Libre open source software development teams," *Journal of Database Management*, vol. 19, no. 2, pp. 1–30, Apr. 2008, bibtext: Crowston2008.
- [62] S.-Y. T. Lee, H.-W. Kim, and S. Gupta, "Measuring open source software success," *Omega*, vol. 37, no. 2, pp. 426–438, 2009.
- [63] C. Subramaniam, R. Sen, and M. L. Nelson, "Determinants of open source software project success: A longitudinal study," *Decision Support Systems*, vol. 46, no. 2, pp. 576–585, 2009.
- [64] M. Soto and M. Ciolkowski, "The QualOSS open source assessment model measuring the performance of open source communities," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, Oct. 2009, pp. 498–501.
- [65] B. D. Sethanandha, B. Massey, and W. Jones, "Managing open source contributions for software project sustainability," in *PICMET 2010 Technology Management For Global Economic Growth*. IEEE, 2010, pp. 1–9.
- [66] M. J. Bietz, T. Ferro, and C. P. Lee, "Sustaining the development of cyberinfrastructure: an organization adapting to change," p. 10, 2012.
- [67] C. M. Schweik, "Sustainability in open source software commons: Lessons learned from an empirical study of SourceForge projects," *Technology Innovation Management Review*, vol. 3, no. 1, 2013.
- [68] L. Nyman and J. Lindman, "Code forking, governance, and sustainability in open source software," *Technology Innovation Management Review*, vol. 3, no. 1, pp. 7–12, 2013. [Online]. Available: <https://timreview.ca/article/644>
- [69] S. Daniel, R. Agarwal, and K. J. Stewart, "The effects of diversity in global, distributed collectives: A study of open source project success," *Information Systems Research*, vol. 24, no. 2, pp. 312–333, 2013.
- [70] A. Head, "Social health cues developers use when choosing open source packages," in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE 2016. New York, NY, USA: ACM, 2016, pp. 1133–1135. [Online]. Available: <http://doi.acm.org/10.1145/2950290.2983973>
- [71] I. R. Viseur, "Open Concentration Index: Measure of Market Concentration in Open Source Industry," in *Proceedings of the 12th International Symposium on Open Collaboration*, ser. OpenSym '16. New York, NY, USA: ACM, 2016, pp. 6:1–6:4. [Online]. Available: <http://doi.acm.org/10.1145/2957792.2957796>



- [72] J. Terrell, A. Kofink, J. Middleton, C. Rainear, E. Murphy-Hill, C. Parnin, and J. Stallings, "Gender differences and bias in open source: pull request acceptance of women versus men," *PeerJ Computer Science*, vol. 3, p. e111, May 2017. [Online]. Available: <https://peerj.com/articles/cs-111>
- [73] D. Izquierdo-Cortazar, J. M. Gonzalez-Barahona, S. Duenas, and G. Robles, "Towards Automated Quality Models for Software Development Communities: The QualOSS and FLOSSMetrics Case," in *2010 Seventh International Conference on the Quality of Information and Communications Technology*, Sep. 2010, pp. 364–369.
- [74] V. Midha and P. Palvia, "Factors affecting the success of Open Source Software," *Journal of Systems and Software*, vol. 85, no. 4, pp. 895–905, 2012.
- [75] L. Laffan, "A new way of measuring openness: The open governance index," *Technology Innovation Management Review*, pp. 18–24, 2012.
- [76] R. Sen, S. S. Singh, and S. Borle, "Open source software success: Measures and analysis," *Decision Support Systems*, vol. 52, no. 2, pp. 364–372, 2012.
- [77] J. P. Spradley, *Participant Observation*. Waveland Press, Feb. 2016, google-Books-ID: q7DICwAAQBAJ.
- [78] A. H. Van de Ven, *Engaged scholarship: A guide for organizational and social research*. Oxford University Press on Demand, 2007.
- [79] J. Fook, "Developing critical reflection as a research method," in *Creative spaces for qualitative researching*. Springer, 2011, pp. 55–64.
- [80] J. van Maanen, *Tales of the field: on writing ethnography*. The University of Chicago Press, 1988, bibtex: Maanen1988.
- [81] K. Krippendorff, *2004 Content Analysis: An Introduction to Its Methodology*. Thousand Oaks, CA: Sage Publications, 2004.
- [82] J. E. Introne and M. Drescher, "Analyzing the flow of knowledge in computer mediated teams," in *Proceedings of the 2013 conference on Computer supported cooperative work*, ser. CSCW '13. New York, NY, USA: ACM, 2013, pp. 341–356. [Online]. Available: <http://doi.acm.org/10.1145/2441776.2441816>
- [83] J. Bacon, *The art of community: Building the new age of participation*. O'Reilly Media, Inc., 2012.
- [84] L. Foundation, "Creating an Open Source Program," 2019. [Online]. Available: <https://www.linuxfoundation.org/resources/open-source-guides/creating-an-open-source-program/>