Open Source Project Health

GEORG J.P. LINK

Georg J.P. Link, PhD, is an open source strategist. During 15 years in open source, Georg experienced the importance of open source project health when the OpenOffice.org community forked the project to start LibreOffice and founded The Document Foundation. This impressive experience inspired Georg's PhD research focus. Today, Georg's mission is to help open source projects, communities, and organizations become more professional in their use of project health metrics and analytics. Georg cofounded the Linux Foundation CHAOSS Project. As the Director of Sales at Bitergia, Georg helps organizations and communities with adopting CHAOSS metrics and technology. In his spare time, Georg enjoys reading fiction and hot-air ballooning. Find Georg online at https://georg.link/ and on Twitter: @GeorgLink, email: georg@chaoss.community

Open source project health describes the potential of projects to continue developing and maintaining quality software, an issue that has long been overlooked. Recently, open source software failures have negatively affected millions of people (e.g., OpenSSL, Equifax), raising the question about the health of open source projects that develop these critically important pieces of software. Measuring and determining the health of open source projects that develop and maintain open source software is a difficult task and has been hard to do well. In this article, I describe issues that make open source project health difficult to measure and what the CHAOSS project has been doing to help with measuring the health of open source projects.

Failures of Open Source Project Health

Software development is often done piecemeal, relying heavily on existing software libraries. For example, the OpenSSL library provides highly specialized encryption algorithms that require expert cryptography knowledge and makes these features available to any developer. This piecemeal approach to software development is fueled by open source software. Increasingly, software libraries are made available through an open source license which encodes the rights for anyone to use, modify, and share the software for any purpose. This licensing model enables developers to collaborate in software production, avoiding duplicate work and improving the software for the benefit of everyone. But despite all the advantages that open source software brings, there are also challenges.

The challenge I explore in this article is in measuring and understanding the health of open source projects. The absence of traditional software project and market indicators makes understanding open source project health quite difficult. The health of proprietary software projects can be measured by revenue from sales that will support future development for the software. Sales figures are nonexistent, and open source licensing means that open source software can be distributed and used by anyone without paying a license fee. Open source project health needs different metrics. This challenge used to be an academic exercise, but today it has the attention of open source foundations, large corporations, and governments. This is because open source projects are a critical part of our digital infrastructure, empowered by projects like OpenSSL, Linux, and Apache Web Server. Many governments, organizations, and individuals depend on open source projects.

Considering the widespread use of open source software, project health failures can have significant impacts. For example, the Heartbleed vulnerability existed in the open source software library OpenSSL [1]. OpenSSL was used by most web servers to secure Internet traffic. Heartbleed allowed a malicious user to get sensitive information from a server, endangering the data of millions of Internet users. This vulnerability was introduced in 2012 and publicly disclosed in 2014. The baffling part of this story is the mismatch between the widespread use of OpenSSL and its very small project community of a few unpaid developers. In hindsight, OpenSSL had poor open source project health, which should have served as a warning signal if only we had paid attention to it.

Heartbleed was a wakeup call to organizations relying on open source software. The Ford Foundation research report *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure* by Nadia Eghbal [2] was very influential in the following conversations. Eghbal had interviewed open source project maintainers and discovered that Heartbleed was merely a very visible open source project health failure while many more open source projects face similar challenges. Some maintainers of open source projects reported suffering burnout from the challenge of securing critical software with little help in their spare time while earning a living in an unrelated job. Several solutions were proposed in response to this realization. For example, the Linux Foundation established the Core Infrastructure Initiative to give money and developer resources to open source projects that were critical for the digital infrastructure but were lacking a healthy project community. Similarly, Mozilla has the Open Source Support (MOSS) program. However, open source project health is more complex than just a matter of lacking financial resources.

OpenSSL's Heartbleed example highlights the need for open source project health to ensure the production of quality open source software. This is not sufficient when users of open source software do not pay attention to changes in the health of open source projects. Equifax, for example, was using the open source software Apache Struts and failed to respond to an update announcement that a vulnerability (CVE-2017-5638) had been fixed in a new version of Struts. Two months after the fix was released, Equifax became subject to a data breach because it was still using a vulnerable and outdated version that hackers exploited [3], and 143 million US consumers were affected. This example highlights that users of open source software have to not only evaluate open source project health once but monitor it continuously and actively for all software and infrastructure components they rely on.

Long-time members of open source projects will tell you that they have developed a sense for open source project health and make decisions based on past experience. However, this sense may not scale to organizations without tools for automation. The open source ecosystem is growing rapidly as more first-time contributors are participating in open source projects. A formalized understanding of how to measure open source project health can transfer this critical knowledge and allow it to be embedded in supporting software.

Measuring Open Source Project Health

Before we can assess open source project health, we need to have clarity on definitions and assumptions. Open source software is at the core of this discussion and is defined as software licensed under an open source license. The Open Source Initiative (https://opensource.org/) is the steward of the Open Source Definition and decides which software licenses are valid open source licenses. The production of open source software is organized in open source projects, which have a technical and a social component. The technical component includes the tools used in software production: source code repositories, issue trackers, mailing lists, CI/CD toolchains, and so on. The social component includes the people involved and how they organize their collaboration: governance, leadership, membership, events, and working groups. Open source community refers to the people involved in an open source project. Just like most people have fingers but unique fingerprints, open source projects have common technical and social components but are not alike. The unique context of each open source project makes it difficult to measure open source project health in a standard and consistent way.

Open source project health is the potential that an open source community will continue developing and maintaining quality software [4]. This assumes that an open source project has the goal of producing software and that the user of the software wants good quality. Because project health is forward looking, an assessment can only speak to the potential and not about a precise probability or likelihood that a community will continue to develop and maintain quality software.

Open source project health can be assessed along three dimensions [5]:

1. Community
2. Code
3. Resources

The *community* dimension captures the idea that open source projects rely on people to contribute. An assessment could look at the diversity of active community members, the size of the community—both contributors and users—and the governance of the community. The *code* dimension captures the idea that open source projects should produce and maintain quality software. An assessment could look at vulnerabilities, code quality, and activity in code review processes. The *resources* dimension captures the idea that open source projects can develop quality software using their own resources, including an infrastructure of specialized hardware, continuous integration systems, testing facilities, and financial resources. An assessment could look at the availability of resources, number of sources providing resources, and how resources are managed within a project. Each of these dimensions focuses on a different aspect of open source project health and can be understood through more metrics than are listed here.

There are two types of data for metrics about open source project health: qualitative and quantitative. Qualitative data can be collected through surveys and interviews with open source community members to understand their perception of a project's health. These valid data collection methods are time-consuming

and are rarely done. Recent examples are the Apache Community Survey 2020 and the OpenStack Gender Diversity Report 2018. Quantitative data is typically easier to process and can be automatically collected. A great source of data about open source projects is the trace data that is created as community members collaborate in the creation of software using computer-mediated technology. This includes the Git log, the mailing list archive, and the issue tracker history. Easy-to-collect metrics include quantifying events, such as the numbers of commits, emails, issues, comments, and functions or lines in the source code. While we know that some metrics are easier to obtain than others, the important question is which metrics are most indicative of open source project health.

To date, there is no canonical set of metrics that are most indicative of open source project health. Several studies analyzed historic metrics and correlated them with the continued existence and development of open source projects. In such a setup, a healthy project was one that was developing and maintaining software at the time of the study, and unhealthy projects had stopped development [6]. However, these studies have failed to determine metrics that will be useful. My work has explored these failures through many conversations with open source practitioners in open source projects, organizations, foundations, and government. The unique ways in which each open source project works influence the interpretation of metrics and have so far thwarted all efforts to develop quality models and definitive open source project health metric guidelines.

Building Shared Understanding of Open Source Project Health

Despite the challenges, many open source communities, open source foundations, organizations, and researchers want to determine the health of open source projects. Many lessons have been learned but numerous attempts at measuring open source project health started from scratch because a common language and tool set was missing. The CHAOSS project is seeking to level the playing field and get everyone a head start for understanding the importance of open source project health and how to determine it.

We founded the CHAOSS project, which is an acronym for Community Health Analytics Open Source Software, at the Linux Foundation in 2017. The mission of CHAOSS is to define metrics and software that can help everyone with measuring open source project health. CHAOSS focuses on the basics, such as describing data sources for collecting data about open source projects, defining metrics that can be calculated from that data, and developing a shared language for talking about open source project health. We provide a central location in the open source ecosystem where anyone who is interested in open source project health can come to learn more, discuss ideas, get feedback, and build on existing solutions.

The CHAOSS project has working groups that define related metrics. The five working groups are Diversity and Inclusion, Evolution, Risk, Value, and Common Metrics. To learn more about the metrics in each working group, visit https://chaoss.community/metrics. The key point here is that these working groups think through a variety of issues related to measuring open source project health. For example, the Common Metrics working group describes lower-level metrics that can be used by other working groups for higher-level metrics. One such metric is Organizational Diversity, which can be used by the Risk working group to assess the risk of a single-vendor dependency or by the Evolution working group to assess the growth, maturity, or decline of organizational engagement. The metric Organizational Diversity describes core challenges around identifying which organizations contributors affiliate with, taking into account job changes, contributors using @gmail and not their work email addresses, or combining identities of contributors who use different usernames and email addresses across different collaboration tools. Through these metric definitions, CHAOSS provides a starting point for anyone interested in determining the health of an open source project.

Open source project health metrics can be divided into leading metrics that change rapidly and lagging metrics that are slower to change. On the one hand, we have a fair amount of influence on leading metrics, such as the number of commits or the time to close issues. Setting a goal to increase a leading metric can directly lead to behavior changes in the community. On the other hand, we cannot easily influence lagging metrics, such as the number of long-term contributors or active users of the software. We have so far not found a relationship between leading and lagging metrics that would allow us to say: if you want to improve open source project health as measured by lagging metric X, you need to focus community activities that change leading metric Y and Z. Maybe such a relationship cannot exist because when setting goals for leading metrics, project members may change their behavior to "game" the metric. Gaming of metrics describes a situation in which behavior is targeted to improve a metric while possibly working against the original goals for which the metric was chosen. An example of this is the Number of Commits metric, which measures developer contributions, but developers can easily split a commit into many smaller commits, creating more managerial overhead instead of producing more contributions. Nevertheless, leading metrics can be used in tactical decisions for improving the health of our projects while lagging metrics may be better for tracking long-term goals, of course, taking into account the context of the project.

The CHAOSS project stays neutral about the interpretation of metrics and what they mean in the determination of open source

project health. This approach to determining open source project health accommodates the fact that metrics are highly context-sensitive, and open source projects have many different contexts. Projects use a different mix of technical and social components. Even when using the same collaboration tools, projects have different patterns of collaboration and expected behaviors. Whereas some projects are run by volunteers, others are run by organizational employees. Some projects have benevolent dictators who make many decisions, while others have committees or governing boards who collectively make decisions. Some projects have CI/CD pipelines and automated tests that facilitate feedback on code contributions, and others rely more on human reviewers. These are just examples of the large variety of contexts that open source projects create and that make it difficult to interpret the meaning of metrics. One approach to overcoming this challenge is to have an expert on an open source project interpret the metrics specific to that context and tell a story of the project's health, informed and supported by metrics. Determining open source project health is therefore storytelling supported by metrics and evidence.

Improving Open Source Project Health

Having an honest assessment of open source project health can inform data-driven decisions. Following this idea, I discuss thoughts on how open source project health can inform different stakeholders. My opinion has been shaped by conversations in the CHAOSS project, the SustainOSS.org community, my PhD research, and my current job at Bitergia.

Open source communities can observe open source project health to learn about themselves. Since metrics are not absolute indicators of project health, changes over time can be helpful to identify when to take action. For example, when core contributors to a project are leaving, then the community may have a project health issue as indicated by a decline in issue tracker activity. Conversely, a spike in issue tracker activity may indicate that more users are asking questions about the software, and engaging them strategically can draw them in to grow the community. However, context matters because a spike in activity could be the result of outside factors. I recently experienced this in the CHAOSS project when the number of issue comments tripled over the course of one month because of students interested in applying for the paid Google Summer of Code mentoring program.

Organizations can observe open source project health to mitigate risk when relying on open source software in their operations and value creation. Project health can also inform organizations' strategic decisions regarding which projects to engage in and how to maximize value extraction from open source software. For example, a decline of development activity in an open source project can be an early indicator of risk, and an organization can dedicate employee time to such a project to make sure it stays maintained and compatible with new technology developments, standards, and regulatory requirements.

Open source foundations can observe open source project health to identify best practices and learn from open source projects that are doing very well to then help other projects achieve similar outcomes. Foundations can also use the same metrics to help themselves by observing, for example, who active members in the open source projects are and recruiting them as new foundation members, strengthening the relationships between open source project members and thereby improving project health. Foundations are stewards of open source projects and need to have early indicators of changes in order to intervene when needed.

Contributors to open source projects can use open source project health to make decisions about which projects they want to be part of and how to have the most impact. Contributors prefer healthy open source projects because they are easier to engage in. For example, an increasing number of contributors pay attention to diversity and inclusion as an important aspect in the community dimension of open source project health. Contributors can learn from healthy open source projects with high code-quality standards and improve their job market opportunities.

Conclusion

Project health is an important topic for many open source stakeholders. Open source projects, organizations, foundations, and contributors need to look for ways to better tell open source project health stories that will help stakeholders form an accurate picture of the health of an open source project. The CHAOSS project is an important collaboration for the creation of a shared understanding of open source project health. It provides many resources to understand open source project health and is a vibrant community where project health is discussed, defined, and measured. The CHAOSS project releases project health standards in the form of metrics definitions, creates tooling to measure metrics, and creates community reports to understand project health. CHAOSScast, the CHAOSS project podcast, is a great source of inspiration because the community shares use cases and experiences that are highly contextualized for specific open source projects. As a member of an open source community, ask yourself these two questions: (1) how healthy is my project? and (2) how can I tell my project health story? Join us in the CHAOSS project so we can help tell your story.

References

[1] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson, "The Matter of Heartbleed," in *Proceedings of the 2014 Internet Measurement Conference (IMC '14),* pp. 475–488: https://doi.org/10.1145/2663716.2663755.

[2] N. Eghbal, *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure* (Ford Foundation, 2016): https://www.fordfoundation.org/media/2976/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure.pdf.

[3] E. Weise and N. Bomey, "Equifax had patch 2 months before hack and didn't install it, security group says," *USA Today,* September 14, 2017: https://www.usatoday.com/story/money/2017/09/14/equifax-identity-theft-hackers-apache-struts/665100001/.

[4] 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 (2015): https://doi.org/10.3127/ajis.v19i0.1221.

[5] G. J. P. Link and M. Germonprez, "Assessing Open Source Project Health," in *Proceedings of the 24th Americas Conference on Information Systems (AMCIS 2018)*: http://aisel.aisnet.org/amcis2018/Openness/Presentations/5.

[6] C. M. Schweik and R. C. English, *Internet Success: A Study of Open-Source Software Commons* (MIT Press, 2012).