Assessing Open Source Project Health

Emergent Research Forum (ERF)

Georg J.P. Link
University of Nebraska at Omaha
Omaha, NE, USA
glink@unomaha.edu

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

Abstract

Assessing the health of open source projects is common amongst people who engage in open source projects. This phenomenological study is the beginning of a research program that investigates open source project health. We find a lack of shared understanding of how people assess open source project health. This paper identifies four themes, namely (1) that assessing open source project health is required, (2) that it is built on the activity within a project, (3) that the process of assessing health is not clear nor well understood, and (4) that a deeper understanding of health comes from understanding aspects of community, code, and resources. This paper contributes to open source project health by documenting the universal need for assessing health, uncovering the importance of project activity, identifying the potential benefit of creating a shared understanding and process of assessing health, and confirming theoretical conceptualizations of project health.

Keywords

Open Source Software, Open Source Project Health, Phenomenology, Engaged Field Research.

Introduction

Engaging in open source projects sometimes requires assessing the health of these projects. Maintainers want to demonstrate their projects' health to attract developers and companies (Fogel 2015). Developers want to contribute to projects that are active and make a lasting difference (Head 2016). Companies want to invest resources into open source projects that are reliable and have long-term stability (Ihara et al. 2014). Open source foundations want to support open source projects to help them be healthy (Riehle 2010). A shared need, regardless of the motivation, is to assess and understand open source project health. In academic literature, studies investigated the precursors to project health (Naparat et al. 2015) as well as how to assess project health (Crowston and Howison 2011). The assessment of open source project health is of much interest to practitioners, especially as the open source ecosystem matures and attracts more companies. Practitioner literature gives differing perspectives on how to build and maintain healthy projects, and how to help others assess the health of a project (Bacon 2012; Fogel 2015). Our study takes a new perspective to investigate the commonalities in assessing open source project health. This is important because practitioners who have a unified understanding of assessing open source project health can rely on each other's assessments. In this research-in-progress paper, we document this evolving phenomenon and outline our planned avenues for future research. This leads to our research question:

RQ: How do practitioners assess open source project health?

Related Literature

Open source project health is a project's ability to continue to produce quality software (Naparat et al. 2015). In the beginnings of open source, the question of project health was targeted to understand what motivated developers, who devoted their free time, to participate and build thriving and healthy projects (von Krogh et al. 2012). The increasing involvement of companies in open source projects changed the conversation about open source project health. Today, the assessment of project health involves a look at community governance for (1) community growth, (2) the quality of software, and (3) financial resources that all determine self-sustainability and future existence of open source projects (Arantes and Freire 2011). Following, these three areas of open source project health are considered in turn.

Community growth acknowledges that open source projects are different and that uniform assumptions are inadequate. One example for community growth relates to the socialization process by which members become more involved, starting as a user, becoming a contributor, and finally taking on the responsibilities of a maintainer (Jergensen et al. 2011). Project health can, therefore, be inferred from the ratio of inflow and attrition of members and the relative sizes of each group of members (Crowston and Howison 2011).

Quality of software acknowledges that a healthy project should produce quality software. Community processes and frequency of software releases have been linked to the quality of software (Khomh et al. 2014). Examples of development activities include the number of issues being reported and closed, the number of code commits, or the activity within specific files (Crowston and Howison 2011). Project health can, therefore, be inferred from development activity as it occurs in the communication and collaboration tools.

Financial resources acknowledge that companies can create control imbalances by providing resources (Dahlander and Wallin 2006). For example, perceived company influence can negatively impact voluntary contributions (Spaeth et al. 2015). The health of a project can be improved through the use of open source foundations which are stewards of open source projects and provide financial backing, legal certainty, and administrative support (Riehle 2010). Project health can, therefore, be inferred from the availability of financial resources and the number and types of sources for these financial resources.

Method

Research Design. Phenomenology is a qualitative research approach for investigating the shared experience of a phenomenon (Creswell 2013). The phenomenon in our investigation is the assessment of open source project health. Phenomenology is an appropriate approach because it enables identifying the shared understandings from informants. The philosophical underpinning is that the phenomenon only exists in the experience of people and can only be understood from their descriptions of the phenomenon.

Engaged Scholarship. Our research team is investigating the phenomenon of open source project health as part of a larger research agenda that involves both in-person and online interactions (Kozinets 2015). On an ongoing basis, we documented weekly conference calls where practitioners discuss assessing open source project health. Within the last year alone, we attended and documented four open source conferences at which we presented and conducted focus groups–engaging with more than 100 people. Our engagement informed the research design, helped find key informants, and informed our findings (Van De Ven 2007).

Interviews. From our engagement, we hear many reports from people assessing open source project health. In response, we recruited three key informants, who have many years of work experience with building and maintaining healthy open source projects and assess open source project health before using software. We conducted interviews to provide depth to our understanding gained during our engaged scholarship. We asked informants to describe their use and experience with metrics they consider most important for assessing open source project health. The interview protocol contained eleven open-ended questions. Throughout, we asked informants to consider the phenomenon from different perspectives: (1) from inside a company that uses an external project, (2) from inside a company that started a project that used to be internal to the company, and (3) from inside an open source foundation that supports projects. Our key informants provided rich descriptions of the phenomenon of assessing open source project health.

Data Analysis. The interviews were recorded and transcribed (Creswell 2013) and positioned against our field notes from the engaged scholarship. The first author read the interview transcripts several times and highlighted words and phrases that informed our understanding of how the informants experienced the phenomenon. The first author coded the interviews by writing comments and thoughts into the margins—developing themes. He identified 70 significant statements in the transcripts, copied them into a table, and formulated a meaning based on the themes. He combined the formulated meanings into clusters of meanings which eventually collapsed into the four themes presented in the next section. Again, the analysis was informed and supplemented by our deep open source engagement.

Findings

Theme 1: Assessing open source project health.

The assessment of open source project health is an essential part of informants' jobs. When looking for new software, open source project health should be considered before the software gets evaluated and eliminated if unhealthy. In the assessment, hard rules have to be met by open source projects to be considered healthy, for example, a certain license may be required. However, requirements for a healthy open source project can be relaxed. An open source project that is controlled by a single company would be unhealthy because it can decline accepting contributions. Another reason that the assessment of open source project health can be important is directly related to informants' work. A project maintainer can use open source project health assessment to make decisions for guiding the project she is responsible for and to show her employer that actions have the desired impact. The takeaway from theme 1 is that assessing open source project health is a requirement in many cases.

> *To me, [project health] should have already been considered before we get to the evaluation. We are looking at, let's say, a list of open source software. To me, they should have already been eliminated from that list, if not classified as healthy. (C:105)*[1]

Theme 2: Activity as a proxy for open source project health.

The assessment of open source project health does not depend only on specific metrics but activity in a project serves as a first indicator. Informants looked for projects that seem active before exploring open source project health in more detail. For example, activity is observed in the time since the last contribution and in the daily or weekly cadence of contributions. Activity is also informed by outside sources such as news articles about a project or a Google search. The reason activity plays this major role as a proxy for open source project health is that a project produces activity when fixing bugs, adding features, or responding to user requests. Therefore, activity is a necessary condition for open source project health and it is easier to observe than other indicators. But activity metrics can be misleading because they do not convey information about quality. Activity can serve as a first indicator only, requiring further consideration. The takeaway from theme 2 is that the activity in a project is a proxy for open source project health.

> *I would say [activity is] the first indicator. That's mainly to avoid integrating a project that's not alive and not healthy. (B:172)*

Theme 3: Unclear process of assessing open source project health.

The assessment of open source project health does not have a defined process. Informants struggled with describing their assessment process. Because a number of factors play into the assessment, the process is unstructured, vague, subjective, and relying on intuition. The qualitative nature of assessing open source project health makes it hard to identify specific metrics. Informants also struggled with identifying a singular meaning for metrics. Metrics have to be interpreted in the context of other metrics and factors. For example, the metric 'number of responses to an issue' does not indicate whether someone from the community is helping or whether they are piling onto the issue, which is important to know for the health of the project. The struggle informants had with assessing open source project health is also rooted in how easy it is to make a wrong health judgment as a project changes. Metrics should be looked at over a rolling time window to see how they differ between points in time. The takeaway from theme 3 is that there is no defined process for assessing open source project health.

> *If I'm looking at a very old project that's been around for a long time, I would probably look at things like how code reviews are done and how contributions are vetted. Early on in a project, I think that's less important, and I think, more superficial metrics, such as the number of issues filed, and even just number of contributions sent [is important]. Those are good early indicators of interest, which is paramount to an open source project in its infancy. (B:282)*

Theme 4: Project health matters in community, code, and resources.

The assessment of open source project health has three important areas in which health matters: community, code, and resources. *Community* includes the size of the community, the diversity of the community members, and the governance of the community. The size of the community is the number of active contributors. The diversity of community members includes the affiliation of contributors with different employers and in the ratio of maintainers to contributors. Community governance relates to concerns about processes, including how code reviews are done, how contributions are vetted, or how contributions are dealt with. *Code* includes the activity in collaboration tools. For example, the number of contributions, time since most recent contribution, and daily or weekly cadence of contributions. *Resources* surfaced when we prompted informants to consider the perspective of an open source foundation. Foundations are believed to care about the well-being of an open source project. While a healthy project

[1] We provide one example quote for each theme. This quote starts on line 105 of interview transcript C, or short (C:105).

would be able to fulfill its mission using its own resources, a healthy project may also convince more companies to become members of the foundation which can translate into additional resources (e.g. financial or technical). The takeaway from theme 4 is that open source project health has three important areas, namely community, code, and resources

> *[...] by theme: Metrics to do with issues. Metrics to do with code and pull requests. Metrics to do with what I call community. Metrics to do with maintainer. Metrics to do with releases and downloads. (A:260)*

> *[Foundations] also really want projects that do have corporate backing and ideally multiple corporations backing the project because then they can be convinced to become members of the foundation and give money. (A:396)*

Discussion

This paper documents the prevalence of assessing open source project health. In theme 1—assessing open source project health—the informants' experience of assessing open source project health is clearly driven by the recurring need for open source project health. This contributes to the research on open source project health (Crowston and Howison 2011; Naparat et al. 2015).

This paper finds that activity within an open source project is a necessary but not sufficient indicator of health. In theme 2—activity as a proxy for open source project health—the informants' experience of assessing open source project health demonstrates the tension between the ease of obtaining metrics and the confidence in their informative value. A practical implication is that anyone assessing open source project health has to be careful about drawing immature conclusions from easy to obtain activity metrics. This insight contributes to our understanding of metrics (Crowston and Howison 2011; Naparat et al. 2015) and how open source projects can inform others of their health status (Bacon 2012; Fogel 2015).

This paper identifies a need to develop a practical way for assessing open source project health. In theme 3—unclear process of assessing open source project health—the informants' experience of assessing open source project health provides several considerations that have to be considered. An implication for practice is that everyone involved with open source projects would benefit from a unified understanding of how to assess project health, which contributes to the practitioner literature (Bacon 2012; Fogel 2015).

Finally, this paper confirms that the theoretically developed areas of open source project health (Arantes and Freire 2011; Naparat et al. 2015) are experienced by people who engage in open source projects. Informants described the three areas as captured in theme 4—project health matters in community, code, and resources. Interestingly, software quality did not appear in this study, only code related considerations, which focuses on activity over quality. The implication for practitioners is that this is a starting place for developing a shared understanding and unified process for assessing open source project health (Bacon 2012; Fogel 2015). Separating the discussion around health into these three areas of concern might provide a way to focus efforts and achieve incremental understanding (Arantes and Freire 2011).

Conclusion

In this paper, we analyzed interviews to answer the question: *How do practitioners assess open source project health?* We identified four themes and make four contributions, one related to each theme.

This research-in-progress paper captures and communicates our current understanding of how practitioners assess open source project health. Although we conducted, recorded, and transcribed interviews with key informants for analysis, this paper is also informed by our deep engagement with open source projects and our interactions with more than one hundred people at conferences and on weekly calls.

As part of a larger research project, this paper is a stepping stone to inform our research agenda. Future research in this area will explore open source project health in the context of different theories. One project we are working on will contribute to control theory by identifying how control occurs in open source projects and how open source health metrics may be used to exert control. A second project we are working on will contribute to signaling theory by framing open source project health metrics as signals. A third project explores understanding health through price signals via a new market mechanism that connects users and developers to incentivize high-quality open source development. Throughout our research program, we will continue to contribute to practice by refining metrics for assessing open source project health.

Acknowledgements and Ethics

This project was supported by an Alfred P. Sloan Foundation Digital Technology grant on Open Source Health and Sustainability, 2016-2020.

This project was supported by Mozilla.

This project method was reviewed and approved by our University's Institutional Review Board.

ORCID

Georg J.P. Link orcid.org/0000-0001-6769-7867

Matt Germonprez orcid.org/0000-0003-2326-5901

REFERENCES

Arantes, F. L., and Freire, F. M. P. 2011. "Aspects of an Open Source Software Sustainable Life Cycle," in *IFIP International Conference on Open Source Systems*, Springer, pp. 325–329. (http://link.springer.com/chapter/10.1007/978-3-642-24418-6_26).

Bacon, J. 2012. *The Art of Community: Building the New Age of Participation*, (2nd ed.), Sebastopol, CA: O'Reilly Media.

Creswell, J. W. 2013. *Qualitative Inquiry and Research Design: Choosing among Five Approaches*, (3rd ed.), Los Angeles: SAGE Publications.

Crowston, K., and Howison, J. 2011. "FLOSS Project Effectiveness Measures," in *Successful OSS Project Design and Implementation: Requirements, Tools, Social Designs and Reward Structures*, H. Benbya and N. Belbaly (eds.), Farnham; Burlington, VT: Ashgate Pub, pp. 149–167.

Dahlander, L., and Wallin, M. W. 2006. "A Man on the Inside: Unlocking Communities as Complementary Assets," *Research Policy* (35:8), pp. 1243–1259. (https://doi.org/10.1016/j.respol.2006.09.011).

Fogel, K. 2015. *Producing Open Source Software: How to Run a Successful Free Software Project*. (http://producingoss.com/).

Head, A. 2016. "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*, FSE 2016, New York, NY, USA: ACM, pp. 1133–1135. (https://doi.org/10.1145/2950290.2983973).

Ihara, A., Monden, A., and Matsumoto, K.-I. 2014. *Industry Questions about Open Source Software in Business: Research Directions and Potential Answers*, presented at the 6th International Workshop on Empirical Software Engineering in Practice (IWESEP), Osaka, Japan, November, pp. 55–59. (https://doi.org/10.1109/IWESEP.2014.12).

Jergensen, C., Sarma, A., and Wagstrom, P. 2011. "The Onion Patch: Migration in Open Source Ecosystems," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ESEC/FSE '11, Szeged, Hungary: ACM, pp. 70–80. (https://doi.org/10.1145/2025113.2025127).

Khomh, F., Adams, B., Dhaliwal, T., and Zou, Y. 2014. "Understanding the Impact of Rapid Releases on Software Quality," *Empirical Software Engineering* (20:2), pp. 336–373. (https://doi.org/10.1007/s10664-014-9308-x).

Kozinets, R. V. 2015. *Netnography: Redefined*, SAGE Publications.

von Krogh, G., Haefliger, S., Spaeth, S., and Wallin, M. W. 2012. "Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development," *MIS Quarterly* (36:2), pp. 649–676.

Naparat, D., Finnegan, P., and Cahalane, M. 2015. "Healthy Community and Healthy Commons: 'Opensourcing' as a Sustainable Model of Software Production," *Australasian Journal of Information Systems* (19:0). (https://doi.org/10.3127/ajis.v19i0.1221).

Riehle, D. 2010. "The Economic Case for Open Source Foundations," *Computer* (43:1), pp. 86–90. (https://doi.org/10.1109/MC.2010.24).

Spaeth, S., Krogh, G. von, and He, F. 2015. "Research Note—perceived Firm Attributes and Intrinsic Motivation in Sponsored Open Source Software Projects," *Information Systems Research* (26:1), pp. 224–237. (https://doi.org/10.1287/isre.2014.0539).

Van De Ven, A. H. 2007. *Engaged Scholarship: A Guide for Organizational and Social Research*, Oxford University Press.