https://public.etherpad-mozilla.org
Morning OSS

Identifying Health and Sustainability Issues Relevant to Your Context

What are the categories of sustainability?
1. Project growth (early phase)
2. Maintainers with project knowledge continue to contribute

What are the indicators of a project likely to be sustainable?
1. Welcoming new contributors
2. Sufficient governance
   2.1. What is sufficient?

Can I use this open source metric to predict whether this project can be shipping quality software in time X. Maybe a project with no gov

What patterns of prior growth, maturity and decline exist? There are multiple paths to success ...
1. Categories : Perhaps 10 or 15
2. An outcome variable: Perhaps "likelihood of existing in 5 years"
3. Pattern examples (not every project is the same: organizational configurations/genres/archetypes/receipes) [Perhaps "genre"]
   3.1. A highly committed core
   3.2. Constant flow of new contriburors
4. Utility
5. Availability

Examples/Contrast:
1. Flask. You just want stable and useful releases.
2. For accounting packages you want growth because nobody has a full featured open source accounting package yet.

Project users and sponsors have goals. The project leaders and maintainers need to be able to know if they are meeting the goals of th
1. Projects are intersections of participants with overlapping goals.
2. AVOID: One grand unified open source software dashboard.
3. Example: The Enterprise chicken plucking framework at Mozilla: Which metrics matter in this case? ... Open source contributors are r

Projects have the resources available to accomplish the needed work to have the software continue to be useful.

Factors
- Approachability (Actionable transparency)
    - Documentation
    - Welcoming
    - A place to ask questions, and there one place
    - example: Is it clear where and how to add a patch?
    - The more visibility of processes, the better ... the more "actionable transparency" ... Hard to measure with trace data, but you can d
        - How long did it take to identify and fix a bug on the project
        - A "contributability study" ...
    - "Actionable transparency testing" ...
- What are the distinctions between different levels and foci of individual contributor motivation ...

Goals:

Understand and shape/guide open source ecosystem

Identify projects that are at sustainability risk (combo of importance, centrality in dependency structure, and health)

Identify inflection points of growth in use and see if it is matched by growth in contribution.

Predict whether an open source software project will continue to produce quality software.

Which fast-moving, automatable metrics are best correlated to, or predictive of, harder to measure impact and contributor reward metric

Choose appropriate measurements to improve the sustainability of an open source project

What factors/metrics should I be paying attention to during various phase of the project lifecycle? (this could either be quantifiable and r

Are there processes or patterns of work behavior that are related to sustainability? Success? Both?

Identify contributor (and user) migration to potentially competing projects/ecosystems (e.g., Rails to Flask?)

WIll this project be around in the next 5, 10, 15 years?

Align what people are motivated to do and what people need to get done - balance the value of users and developers.

Questions:

What measures are not visible in the trace data?
- users that experienced issues but didn't report them
- potential contributors that considered contributing but didn't
- build speed, first contribution friction.
- Is the project a good place to realize people's goals?

What are the trade offs between generativity and achievement of strategic aims?
e.g. Open Stack attemped road mapping, shaking out what needs to be accomplished together.

Who does refactoring?

When is it necessary to bring new resources to a project? What resources are necessary under given conditions?

Are the technical practices that make recognition of a contribution more difficult? "Squash merge: It sounds good, but its a complex ende

Metrics:
Must be aligned with the goals of the users and sponsors to some degree ... perfect alignment may not be desirable or good.

Do tasks that were considered desirable but hard become easier to accomplish over time ("productive deferral").

How easy is it to find a piece of code where the required change will be implemented ... ??

Is there any connection between ease of contribution and dead code? (Does dead code slow down comprehension of codebase).
- If dead code exists, then its likely that newbies first contribution attempts could be on dead code
- Software engineering metrics:
    - Code coverage
    - Presence of dead code.

Additional resources:

http://urssi.us/resources/

Howison, J., Deelman, E., McLennan, M. J., Silva, R. F. da, & Herbsleb, J. D. (2015). Understanding the scientific software ecosyst
https://doi.org/10.1093/reseval/rvv014

https://www.feverbee.com/how-to-build-an-online-community/

Kraut, R. E., & Resnick, P. (2012). Building Successful Online Communities: Evidence-Based Social Design. The MIT Press.

https://github.com/chaoss/metrics