

OSS Health Index

Survey Results and Breakouts

IQSS team

Chase Harrison, Julian Gautier, Derek Murphy, Mercè Crosas

The Open Source Software (OSS) Health Index Project

- Develop a framework for evaluating OSS applications and their communities
- Propose an approach for building scales to quantify an OSS health and sustainability
- Scales will be ideally based on existing data
- A focus on OSS used in academia for research and libraries
- By good “Health” we initially mean:
 - A project that flourishes, helps conduct excellent research, is useful to its community, will last, ...

The Human Flourishing Program

<https://hfh.fas.harvard.edu/measuring-flourishing>

Pre-workshop Questionnaire

- Goal to help frame content and stimulate thoughts
- Items designed to be preliminary -- subject to discussion today
- Items created by IQSS team, reviewed by Sean Goggins & Matt Germonprez (CHAOSS)
- 33 Factors in 4 categories, based partially on CHAOSS and LYRASIS projects
- Additional comments as open-ended responses
- Administered to panel of experts (workshop participants and additional)
- Computer-assisted internet data collection
- 23 responses out of 37

Community: 10 Factors

- Diversity of professional backgrounds included in project community
- Ease of adding code for new contributors
- Effectiveness in quickly resolving software issues
- Frequency of project contributions
- Friendliness of project community toward newcomers
- Inclusivity of project community
- Number of active individual users of the project software
- Number of organizations actively using the project software
- Recognition and rewards for contributors
- Shared goals among community members

Technical Factors and User Experience: 11 Factors

- Comprehensiveness of software testing of source code
- Ease of project software deployment
- Ease of using project software with other software
- Formal quality assurance models used in source code development
- Frequency of bug reporting in source code
- Frequency of new project software releases
- Number of other software components (e.g. libraries) integrated with this project
- Portability of project software across environments
- Reported security vulnerabilities in source code
- Usability of project software
- Use of discipline-specific standards relevant to the software project's users

Communication: 10 Factors

- Appropriateness of project's software license(s)
- Availability of explicitly defined software citation
- Ease of accessing source code
- Effectiveness of communication between project and users
- Presence of community mailing list(s)
- Regularity of community events
- Reputation of software in relevant professional communities
- Responsiveness of technical support
- Thoroughness of project documentation
- Transparency of project roadmap

Governance:

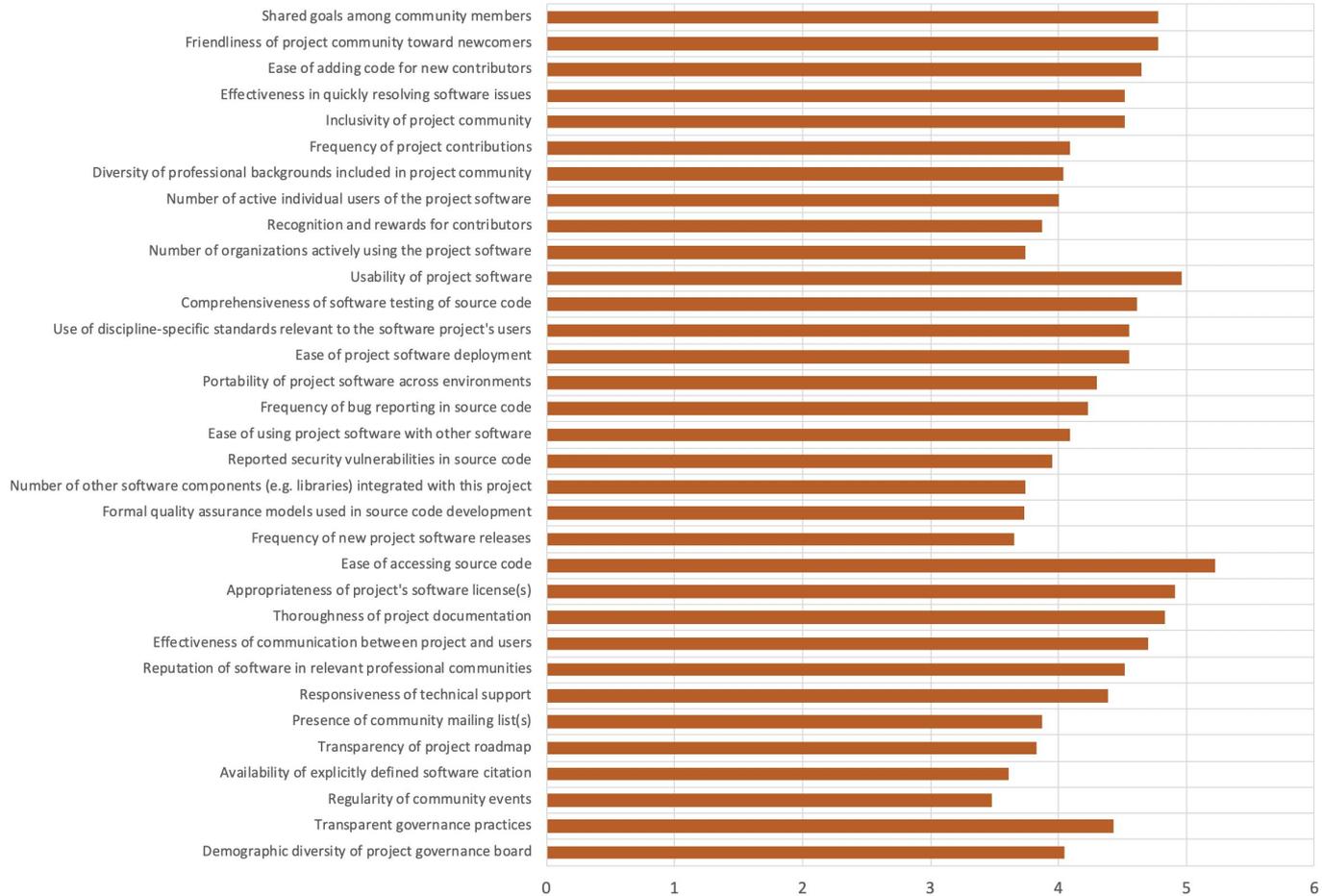
Two unidimensional factors:

- Demographic diversity of project governance board
- Transparent governance practices

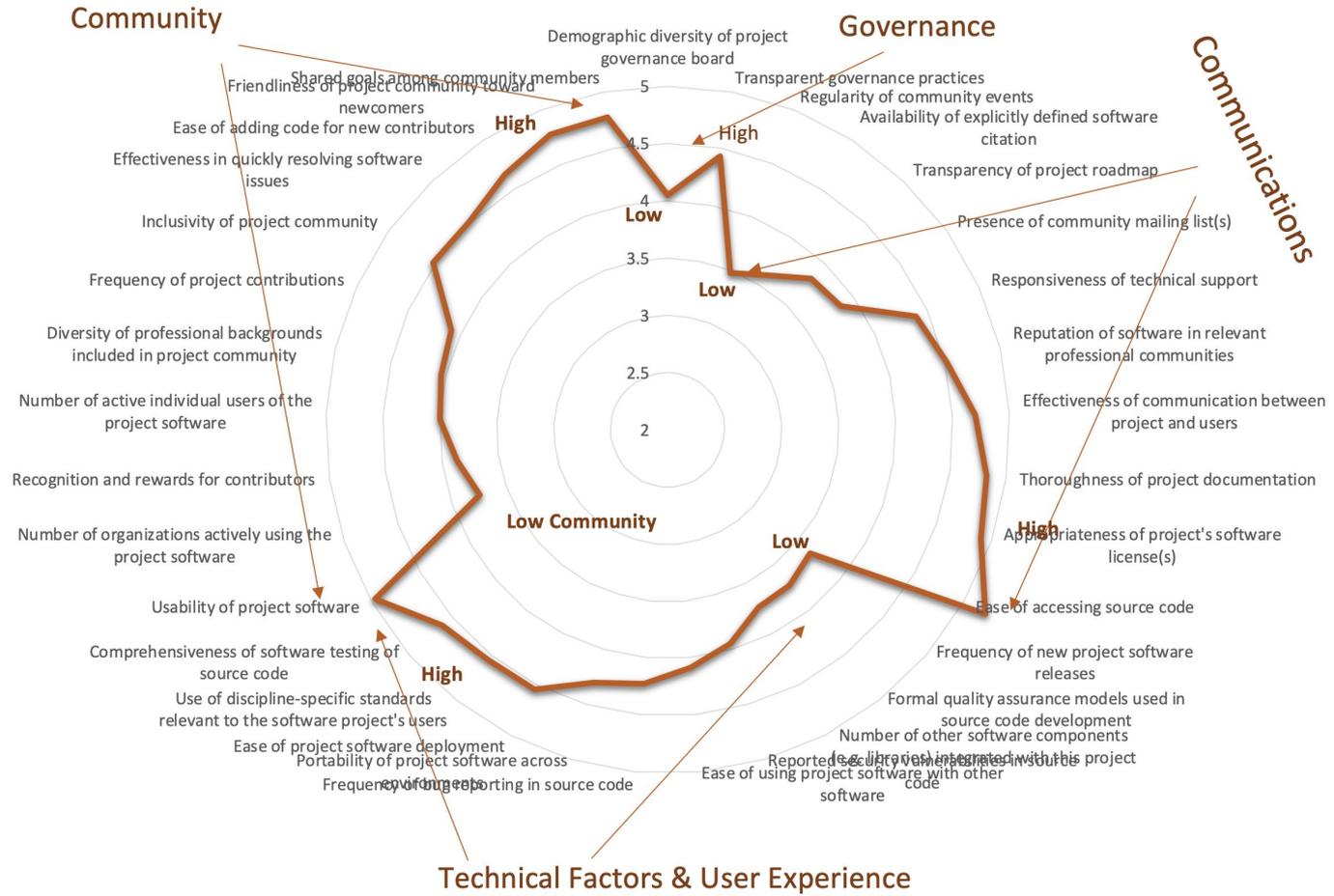
Plus :

- Centralized vs. decentralized project control (two-dimensional scale)

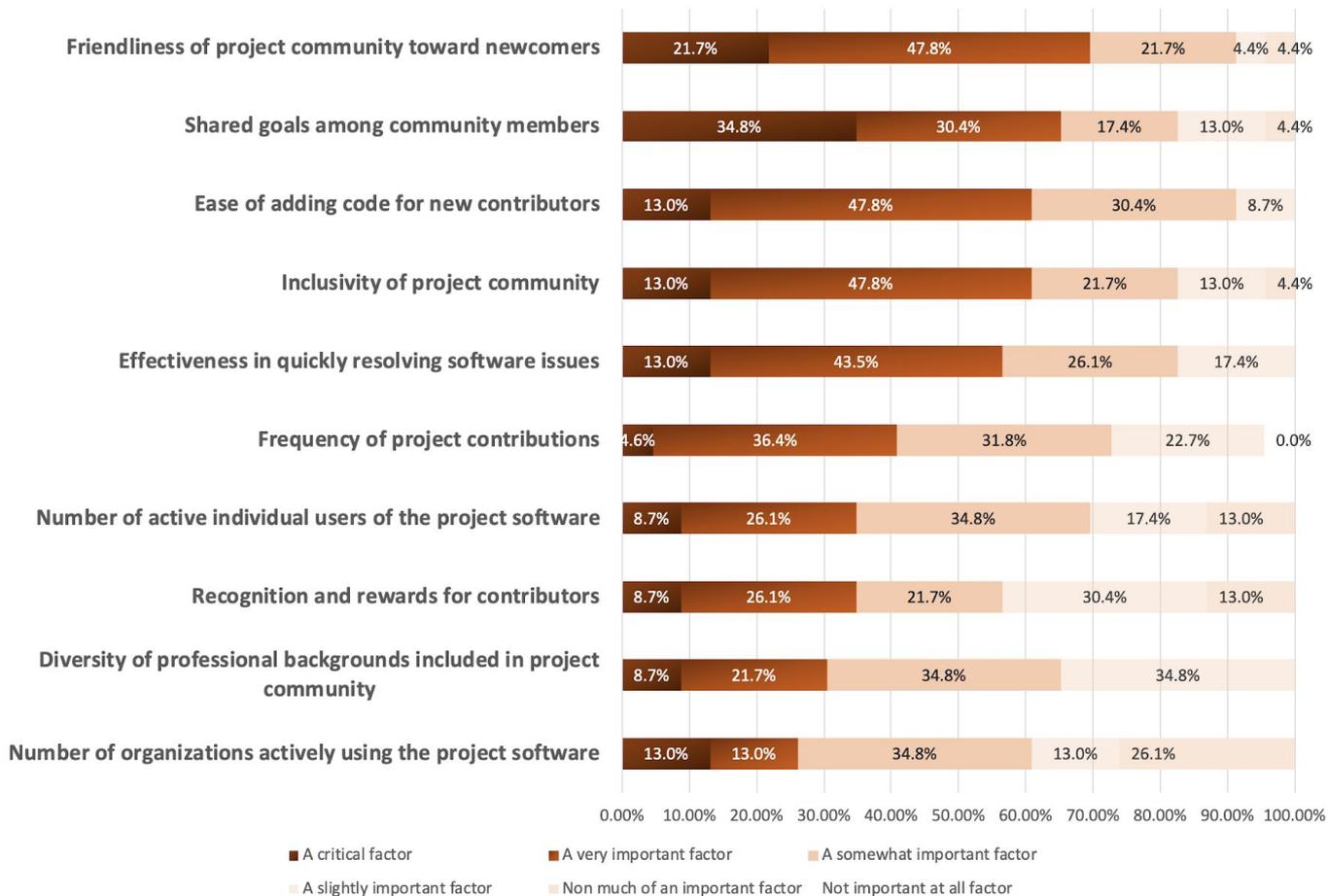
Mean Rating for Factors



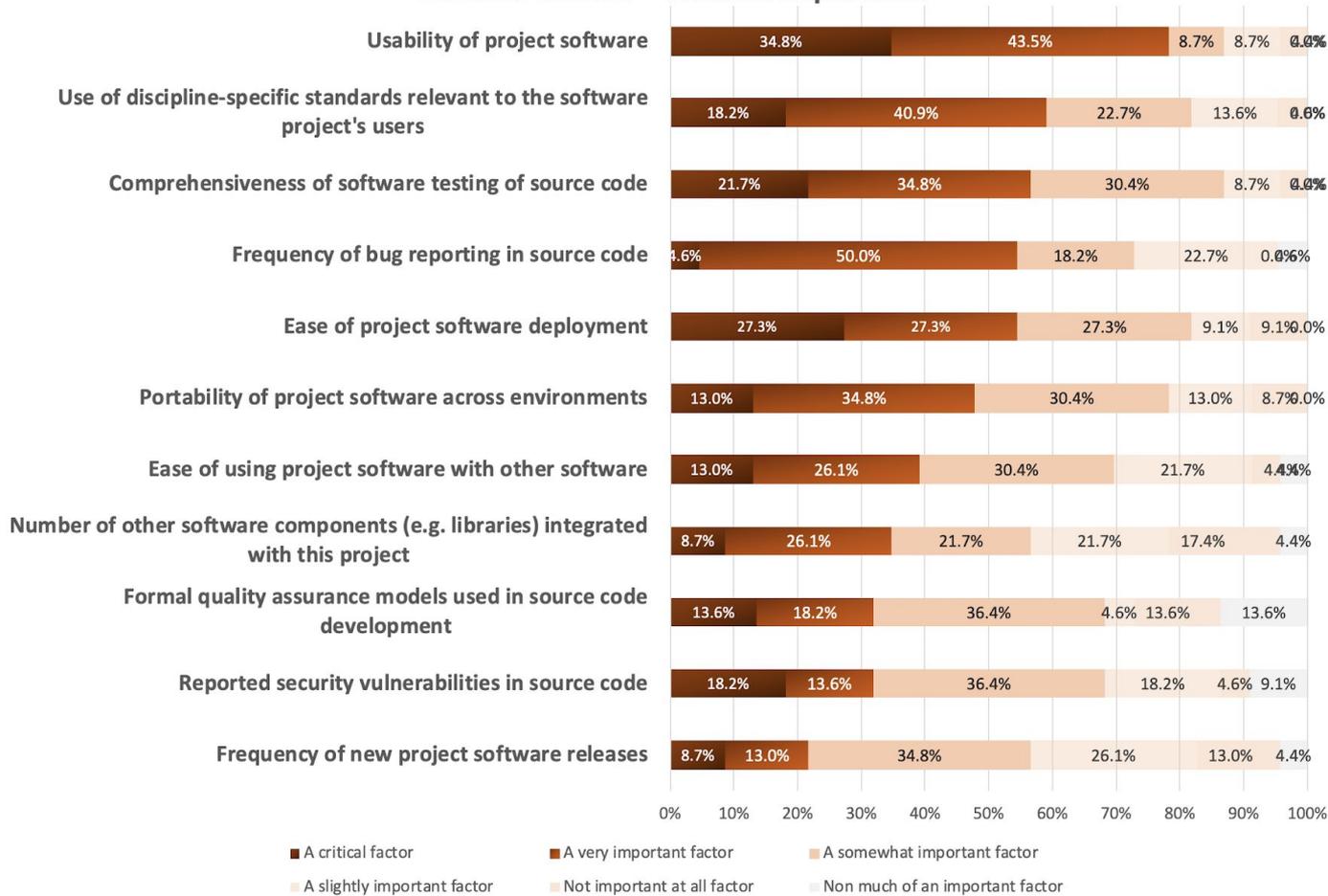
Factor Strength Map



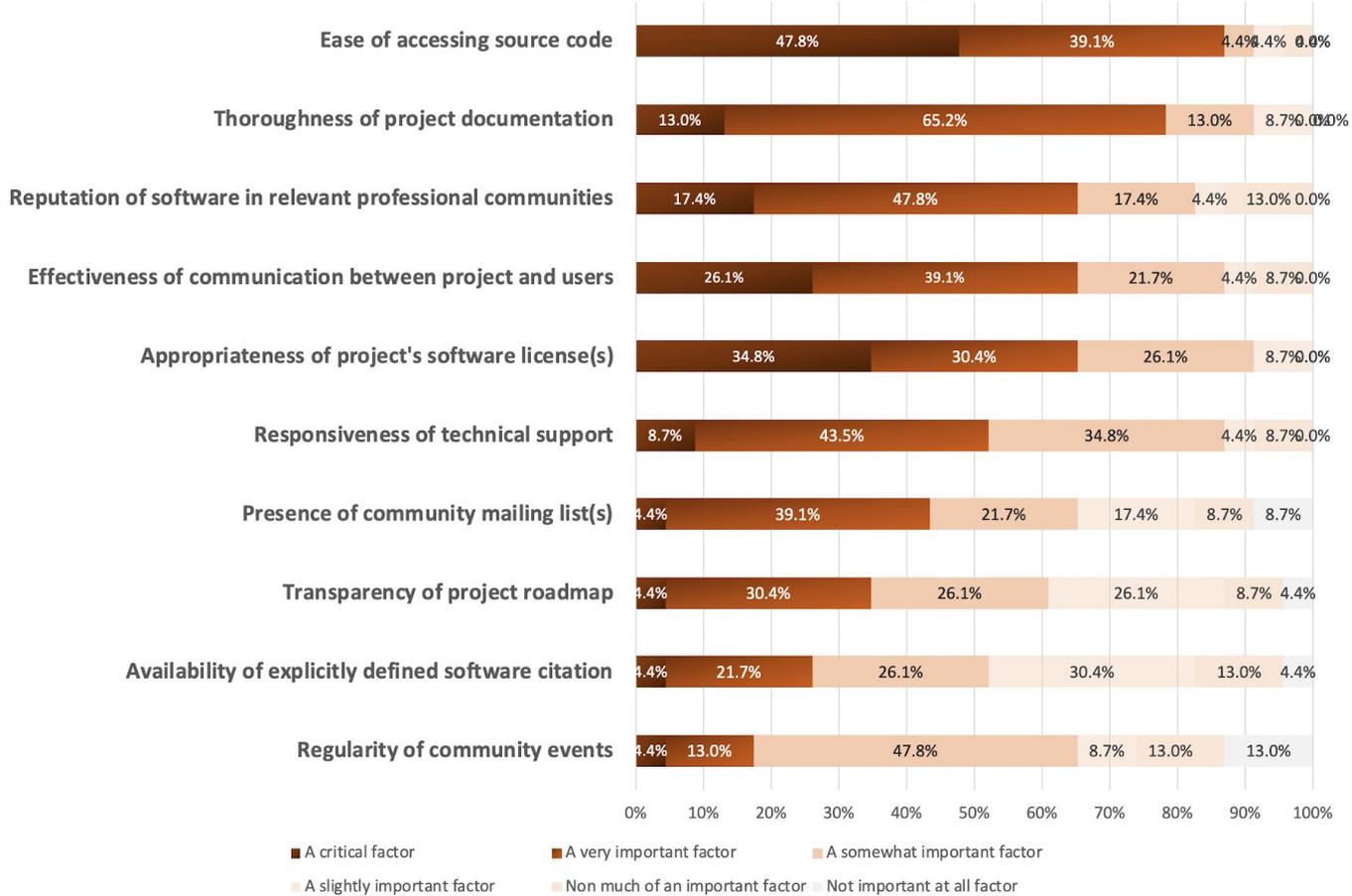
Community Factors -- Relative Importance



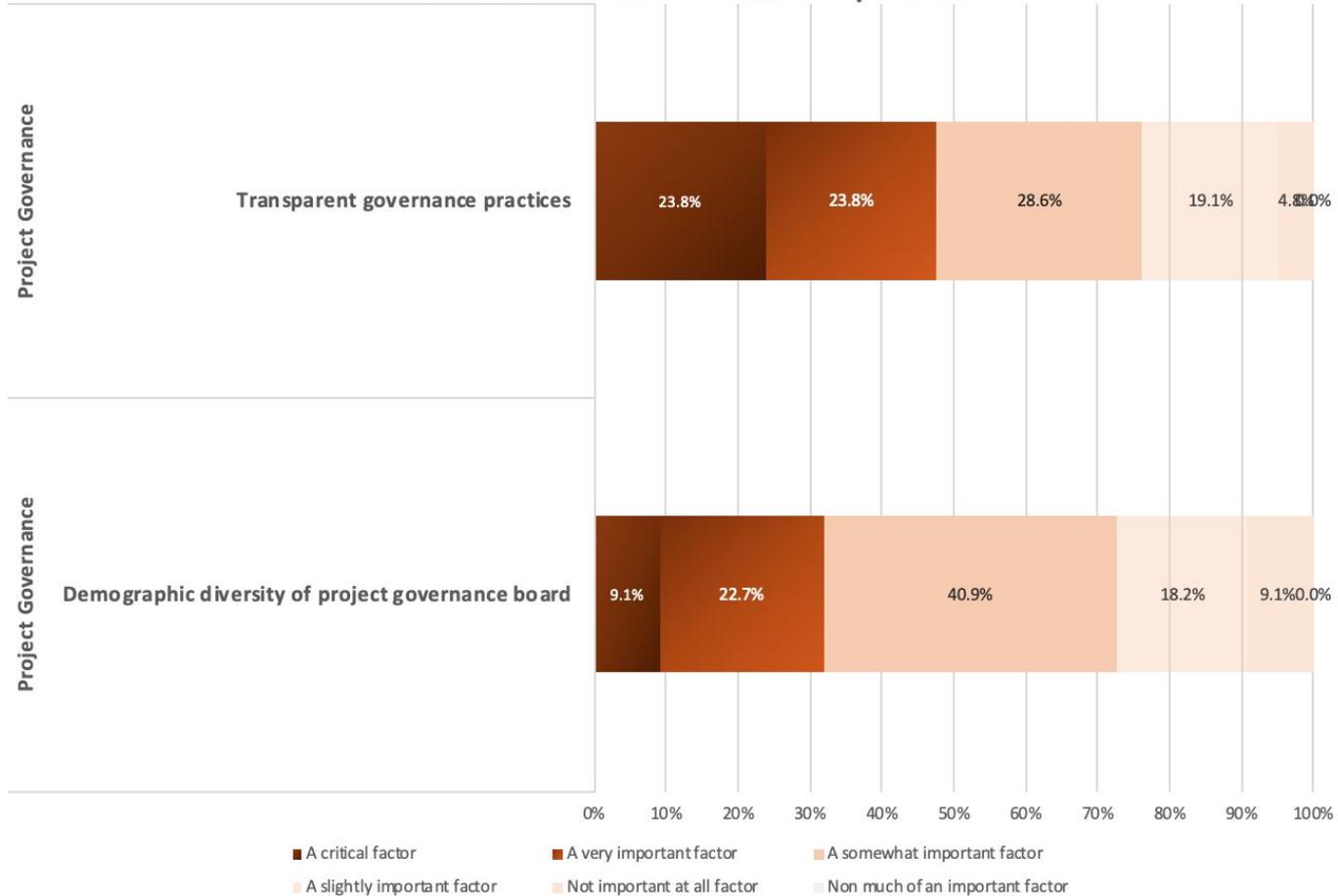
Technical Factors -- Relative Importance



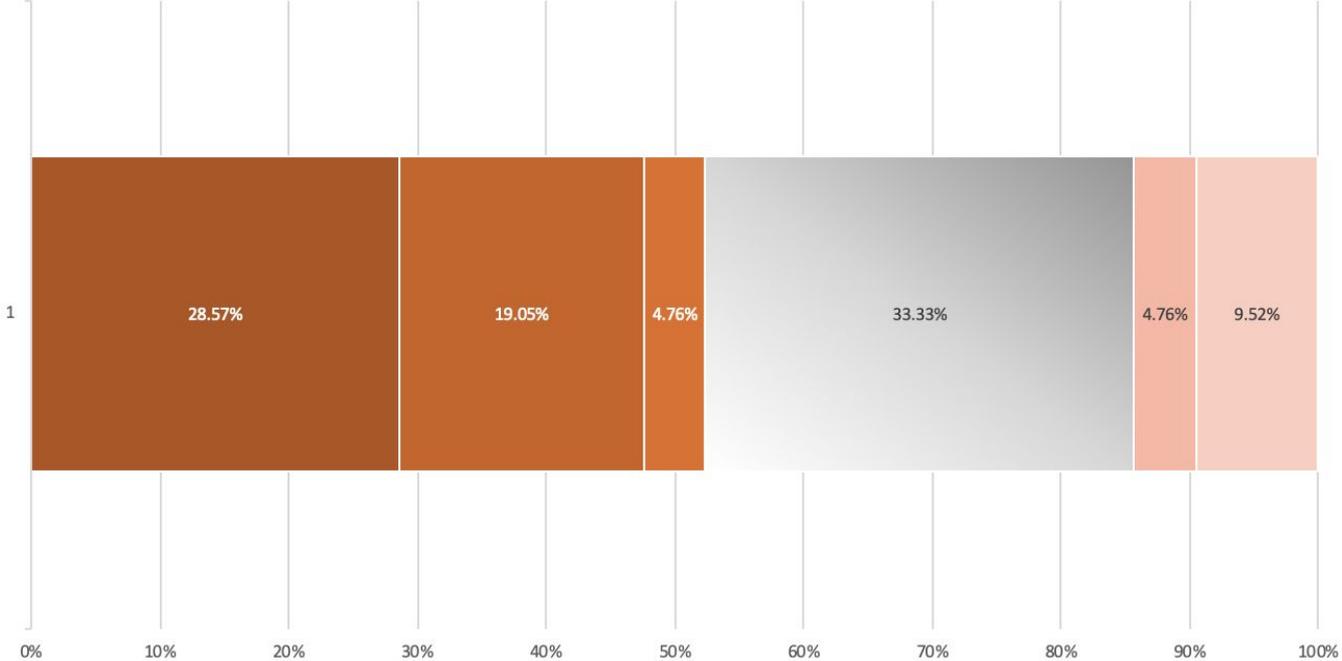
Communications Factors -- Relative Importance



Governance Factors -- Relative Importance



Centralized versus Decentralized Project Control



- Centralized project control is much better than decentralized project control
- Centralized project control is somewhat better than decentralized project control
- Centralized project control is slightly better than decentralized project control
- Centralized and decentralized project control are equally good
- Decentralized project control is slightly better than centralized project control
- Decentralized project control is somewhat better than centralized project control
- Decentralized project control is much better than centralized project control

Comments Summary

- Survey answers depend on goals of project, types of communities it serves, software lifecycle and scale, and how these things naturally change over time.
- What do we mean by "health"?
- Does calling these "projects" imply that they have shorter-term goals? Is "program" a better term?
- What do we mean by "open source"?
- Does the intent of some projects clash with our ideals of open source (e.g. for-profit or self-serving communities/organizations marketing their products as "open source")?
- Questions seem to assume that open-source projects should be community projects.
- What do we mean by "friendliness"?
- Questions about funding models seem to be missing. Are they important?
- Some questions could be phrased more clearly, but I think they are good overall.

Additional factors survey takers wrote in

- Code of conduct exists
- Tutorials
- Community members' sense of engagement with program and community
- Contribution is fun
- Does the project have a dedicated maintainer?
- Ability to pay staff/contractors
- periodic triage meetings
- sustainability prospects
- Contributor Guide exists
- The project has solid branding, and uses it (Twitter, stickers, etc.)
- Maintainers are nice
- How long lived is the project?
- Ability to pay for admin support
- Evidence of Gender Diversity in the Core Maintainers
- Documentation is beautiful
- Is the software developed in multiple languages?
- Ability to accept in-kind support from companies
- Automated testing
- There is a place for (live) open discussion
- Is the software used in multiple distinct disciplines?
- Ability to accept donations or earn money in some way

Breakouts

Four Breakouts

1. Community

- a. Moderators: Matt Germonprez, Mercè Crosas
- b. Room: S020 (Belfer Case room)

2. Technical Factors and User Experience:

- a. Moderators: Sean Goggins, Julian Gautier
- b. Room: S003

3. Communications:

- a. Moderators: Ann Whiteside, Derek Murphy
- b. Room: S001

4. Governance & Funding:

- a. Moderators: Laurie Arp, Chas Harrison
- b. Room: S040

Breakout Questions

- Are meaningful factors missing in your category?
- Which factors become more or less important depending on the phase of the project? (Phases based on “It Takes a Village” Guidebook)
 - Phase I: Getting Started
 - Phase II: Growing/ Getting Established
 - Phase III: Stable, but not static
- Which factors become more or less important depending on the type of OSS?
What types of OSS are you considering?
- Are we missing some additional meaningful categories?
- Anything else?