How Can a Bunch of Amateurs Produce Quality Software?

Miklós Koren (CEU, HUN-REN KRTK, CEPR and CESifo)

February 15, 2024[1]

[1] This work was funded by the European Union under the Horizon Europe grant 101061123. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Software is eating the world

The weightless economy

“Software is eating the world.” (Andreessen, 2011)

Open-source software (OSS) is everywhere

Linux, Apache, MySQL, PHP, Python, R, Julia, Android, Firefox, Chrome, etc.
Also included in proprietary software

Two economic puzzles in open source

Why do people work for free?

Altruism, reputation concerns, alternative business models. Sizeable economic literature.

Two economic puzzles in open source

Why do people work for free?

Altruism, reputation concerns, alternative business models. Sizeable economic literature.

How can a bunch of amateurs produce quality software?

Salient features of OSS

Price is zero

Not even that unique.

Salient features of OSS

Price is zero

Not even that unique.

Scratch your own itch

Developers are often their own first users: grep, TeX, Linux, git, etc.

Free access to source code

"Given enough eyeballs, all bugs are shallow." (Raymond, 1999)

Software quality is only partly observable

Testing is important.

Based on two studies

Success and geography in the weightless economy: Evidence from open-source software

Békés, Hinz, Koren, and Lohman. 2024.

Bugs 🐞

Koren, Békés, and Hinz. 2024.

Data

GitHub

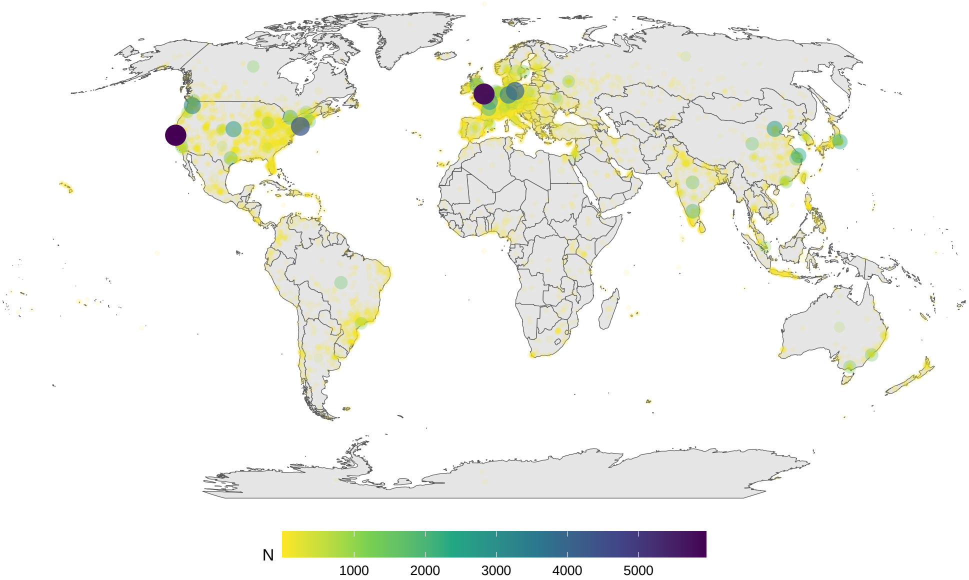
Snapshot of all public repositories on GitHub on 2019-06-01. Six largest languages: JavaScript, Python, Java, Ruby, PHP, and C++. Drop smallest and largest projects. 4.4m projects, 2.7m users. Self-reported location.

libraries.io

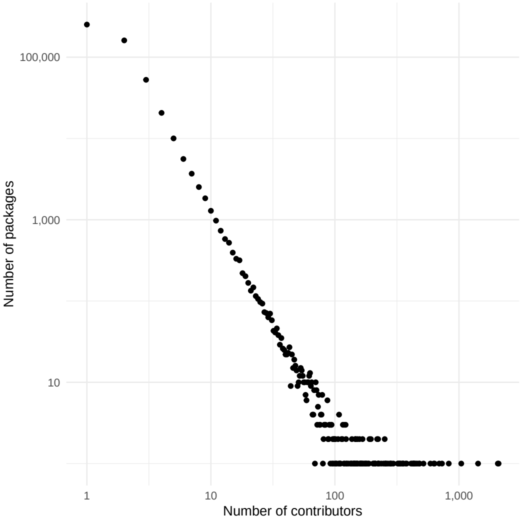
Dependency data for projects on major package managers (npm, PyPI, Maven, RubyGems, etc). Studying npm (JavaScript) today.

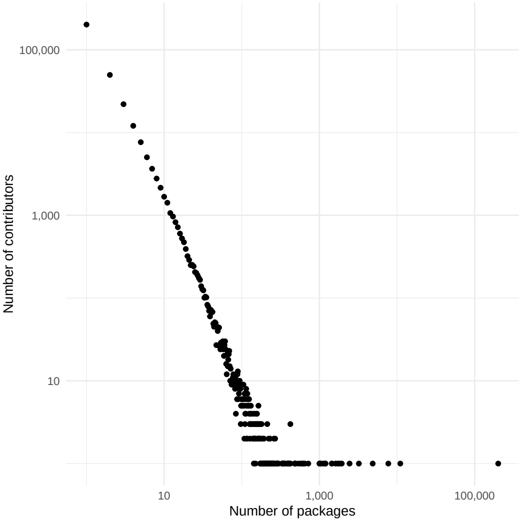
Success and geography in the weightless economy: Evidence from open-source software

Developer density around the globe

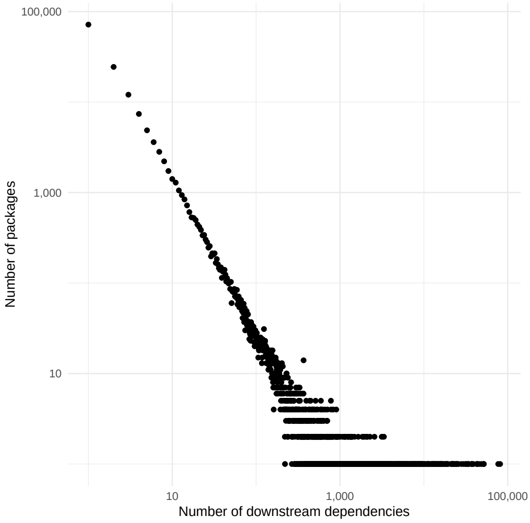


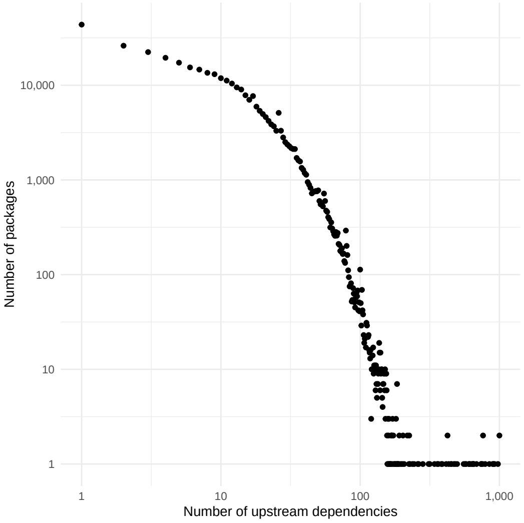
Large variation in number of projects and developers



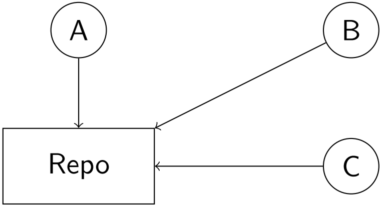


With limits on how many projects one imports

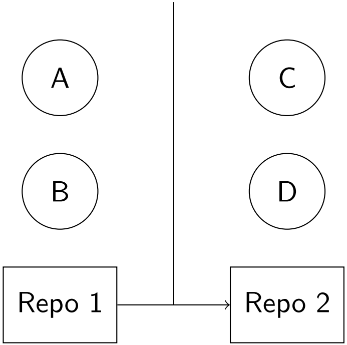




Measuring collalboration and dependencies

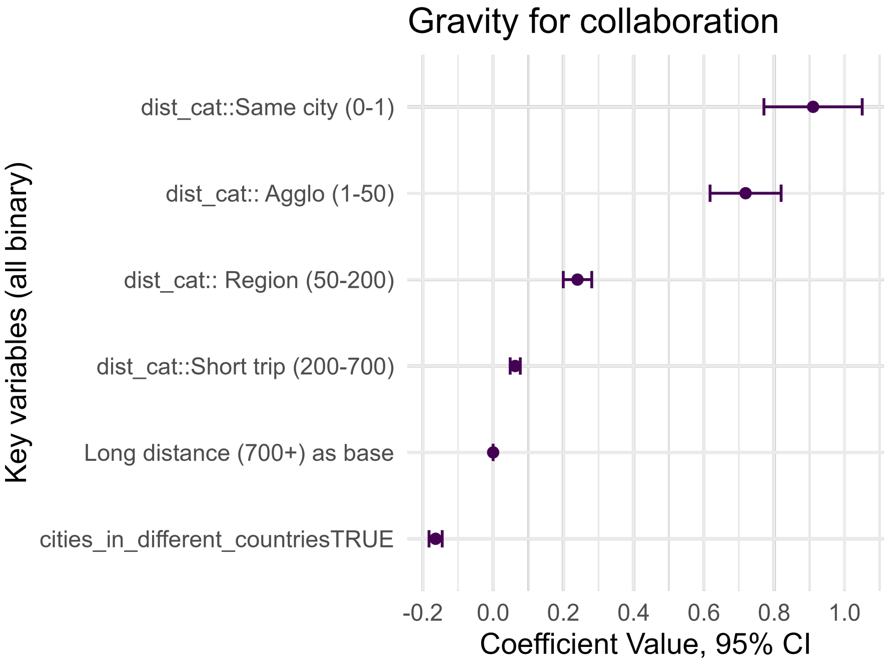


(a) Developers committing to a repository.

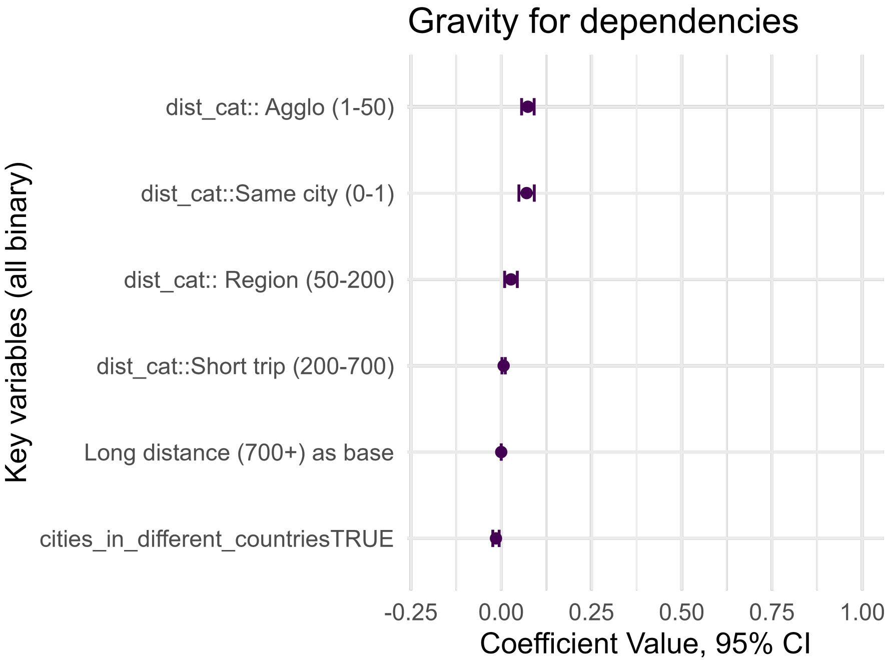


(b) Dependency of repository 1 on repository 2 with the respective developers.

Strong localization of collaboration patterns



No localization of dependencies



Additional results

Organizations help overcome distance. Almost no distance penalty for developers within the same GitHub organization.

Spatially diverse teams are more productive.

Bugs 🐞

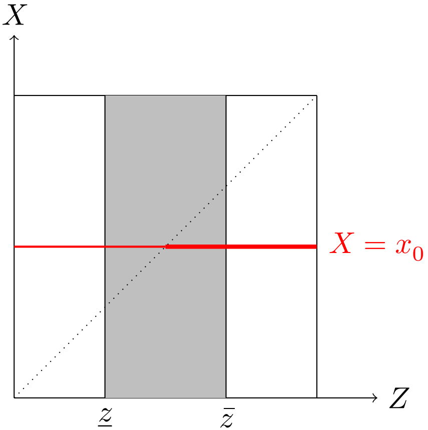
Model

Long-standing question in economics: how does competition affect innovation?

Model the special features of the OSS market.

Market for software quality

- Two-sided market with U users and D developers. Each user meets n developers at random.
- Users have a use case X.
- Developers write code and tests. Software quality is random Z.
- Software only works if $Z > X$.



Competition

Developers compete for users in quality. No price competition.

User selects the best of n software packages. (Platforms → higher n.)

How does quality depend on market structure?

U, D, and n.

Predictions

Standalone projects are limited by developer's own need.

Skilled developers can leverage their quality in projects shared on platforms → contribute more.

Platform projects are tested more than standalone projects.

Larger projects are written by more people

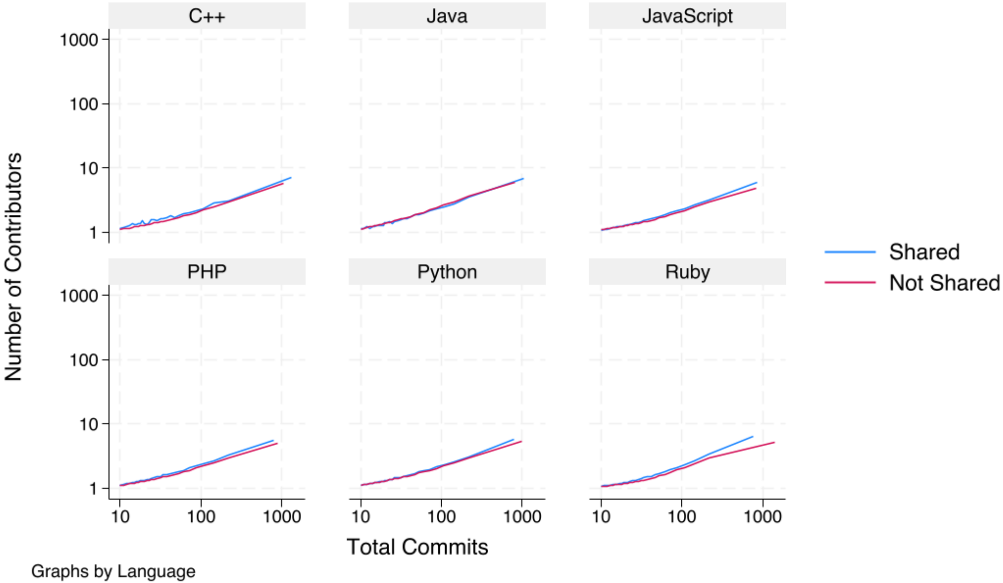


Figure 2: Larger projects by more people

Larger projects are more popular

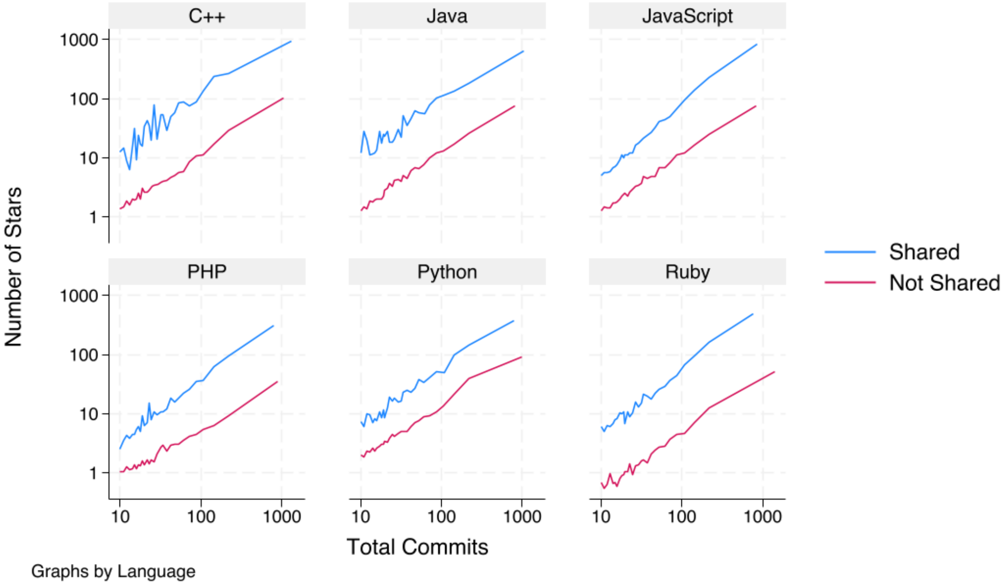


Figure 3: Larger projects are more popular

Larger projects have more bug discovery

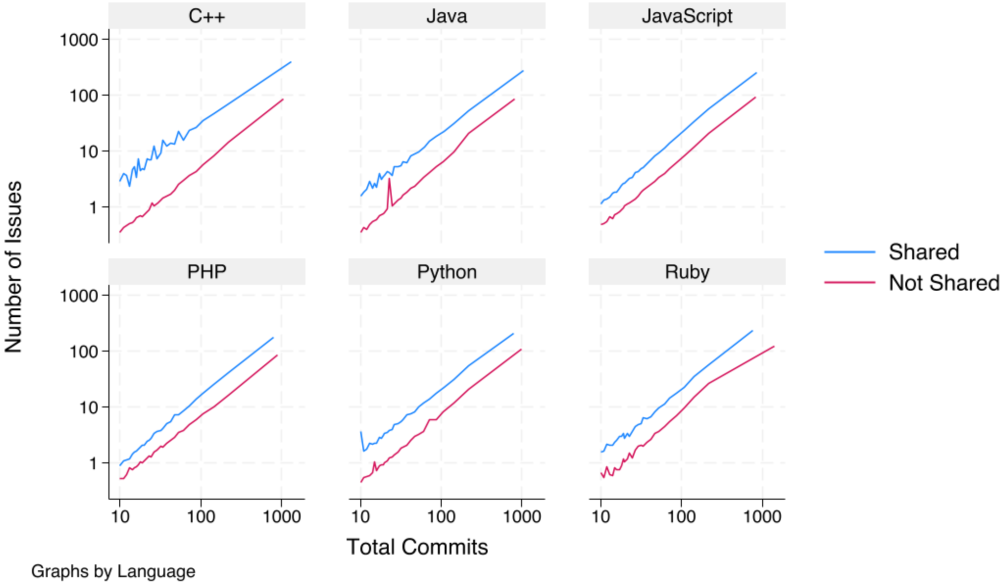


Figure 4: Larger projects have more bug discovery

Larger projects solve a larger share of issues

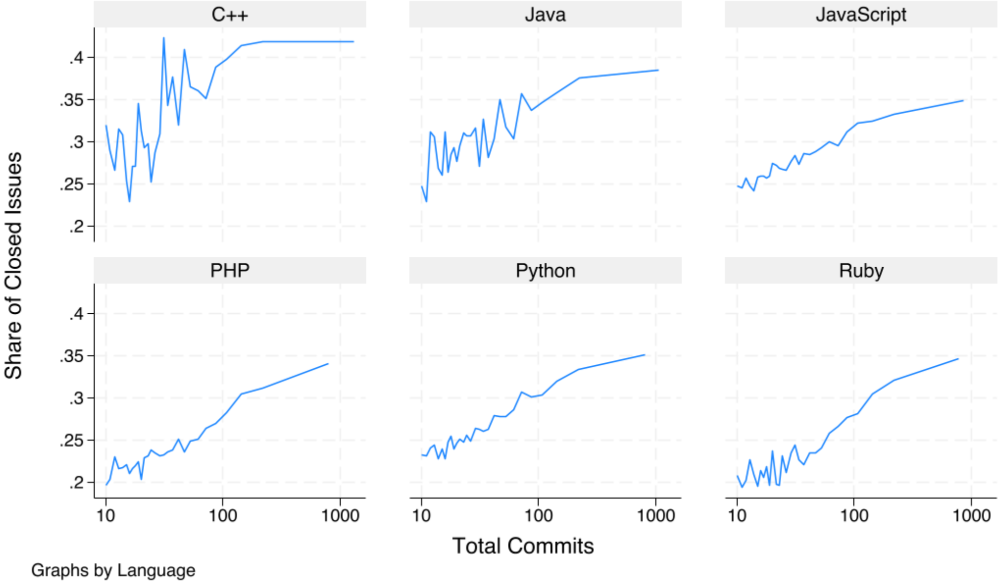


Figure 5: Larger projects solve a larger share of issues

Better developers contribute more to shared projects

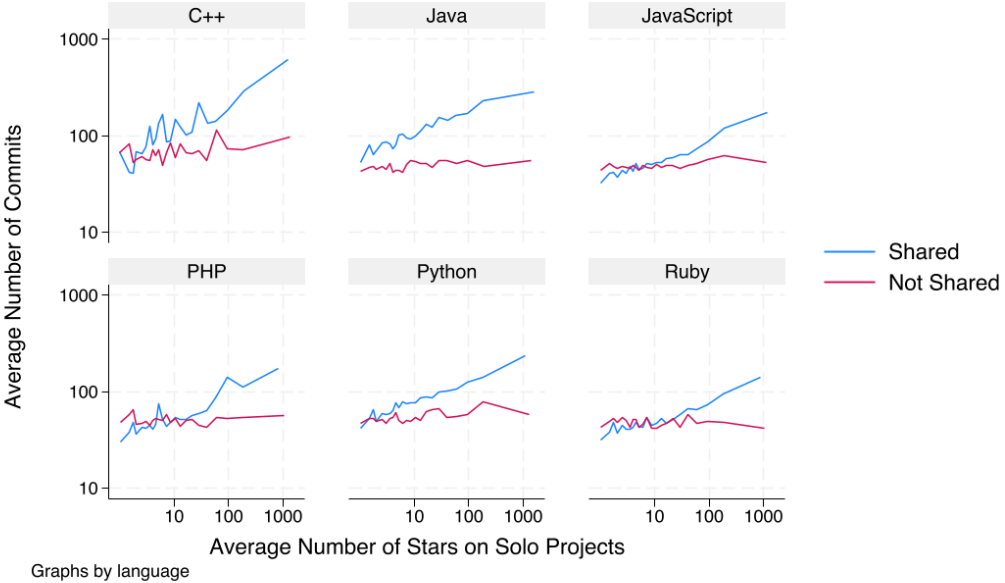


Figure 6: Better developers contribute more to shared projects

Shared projects are better quality

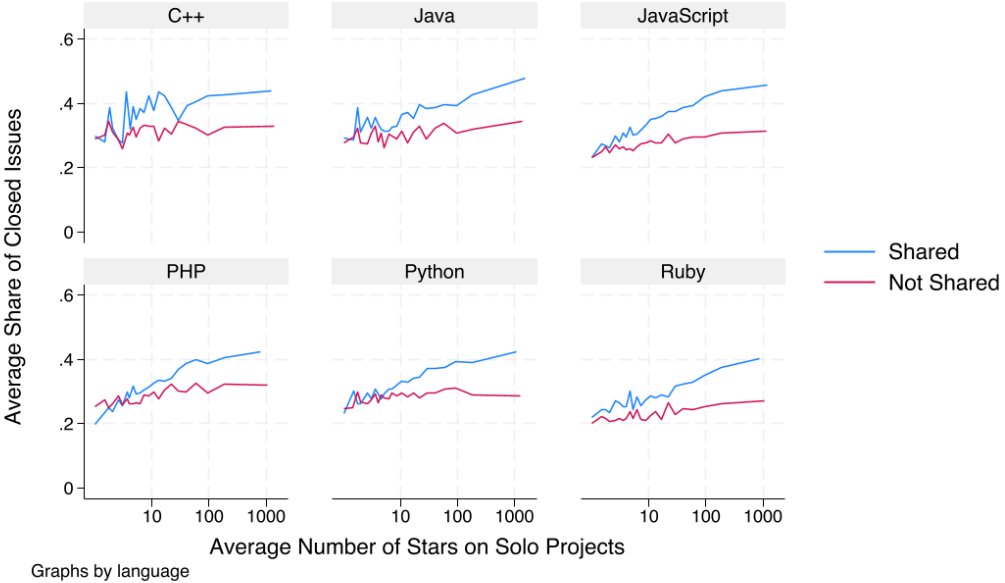


Figure 7: Shared projects are better quality

Next steps

Measure test coverage.

Interaction with users: bug reports, feature requests.

Sorting into collaboration.

Get in touch

@korenmiklos

@gaborbekes

@julianhinz