

## SCS' open source review

SCS is a free and open-source software under MIT license, which means anyone with the necessary skills can contribute to the original project source code. Therefore, the community dynamics around the software greatly influence the software itself and the experience of developers working with it. This chapter will review the open-source quality of this software and, in particular, its community dynamics. After quickly talking about its licensing, we will analyze its community dynamics.

### MIT licensing

The MIT license is one of the most popular permissive licenses. A permissive licensing means little restrictions on how the software can be used, modified, and redistributed. In addition, it allows the software to have high license compatibility, meaning that it can easily be distributed with software under other licensing, including proprietary licenses.

## SCS open source community

### *Hypothesis*

During the experience of this thesis, we had to interact daily with SCS' community. Those would include submitting issues and starting discussions on SCS' source code repository, chatting with the community on the Gitter channel, or Zoom meetings. Through personal experience interacting with SCS' community, we had good support from the community, but mainly from the core developers. Ergo, we will make the following hypothesis and try to gather metrics to validate them.

- For the moment, SCS is built mainly by core developers, and there is not so much contribution from the community
- Most of the tickets answer comes from core developers
- The core developers are dynamically working on the repo
- The developers are quickly answering Pull Requests ( PR ) and tickets

### *Method*

To verify the former hypothesis, we build a series of scripts specially crafted for this chapter . Those scripts take advantage of `mergestat`, an open-source tool to make SQL queries to a git repository that we mainly used to get data from SCS' Github repository in a JSON format. We also query Github API directly when needed.

### *Results*

To interpret the gathered data correctly, we first need to define a few groups of Github users:

- **bots:** Github's bot such as renovate, dependabot and gitter-badger are always removed from the gathered data, as we are only interested in human community dynamics.
- **core-developers:** which consists of the following users: joachimvh, Ruben-Verborgh rubenswork and matthieubosquet<sup>1</sup>. Since Digita ( a European based company promoting Solid and working with SCS ) were highly influencing the results, we decided to create the two following groups:
- **non-core developers:** are defined by all Github users that do not belong to the core developers, including Digita's team.
- **newcomers:** are a subset of the non-core groups, excluding Digita's team.

From the data gathered, we extracted the following information:

- 74% of the Pull Requests ( PR ) are made from the core-developers
- 27 PR made by different non-core developers, totaling 128 new PR ( 1/5 of total PRs)
- 25% of the issues are left unanswered and in particular:
  - 10% of non-core developers left unanswered
  - 2% of newcomers left unanswered
- non-core developers raise 40% of issues
- 73% of the raised issue get a first attention ( a first attention is the first answer in an issue thread )
- core developers wrote 90% of the first attention

### *Discussion*

With the former information, we can validate our hypothesis related to the involvement of the core developers in the community: they answer to most of the issues of newcomers . However, surprisingly, we can see that 20% of the PR comes from non-core developers, which shows a genuine enthusiasm for the software from the community. In conclusion, we can say that the core developers are strongly involved in the community, as they consistently answer issues and questions raised by newcomers. Of course, a community around the software is far from non-existent, but it seems that they are more active in creating issues and submitting PRs rather than solving issues raised by others. It might be because the community lacks expertise since SCS is still a young software.

---

<sup>1</sup><https://github.com/solid/community-server#-license>