Navigation Support in Evolving Open-Source Communities by a Web-Based Dashboard

Anna Hannemann, Kristjan Liiva, and Ralf Klamma

RWTH Aachen University,
Ahornstrasse 55, 52056 Aachen, Germany
{hannemann,liiva,klamma}@dbis.rwth-aachen.de

The co-evolution of communities and systems in open-source software (OSS) projects is an established research topic. There are plenty of different studies of OSS community and system evolution available. However, most of the existing OSS project visualization tools provide source code oriented metrics with little support for communities. At the same time, self-reflection helps OSS community members to understand what is happening within their community. Considering missing community-centered OSS visualizations, we investigated the following research question: Are the OSS communities interested in a visualization platform, which reflects community evolution? If so, what aspects should it reflect?

To answer this research question, we first conducted an online survey within different successful OSS communities. The results of our evaluation showed that there is a great interest in community-centered statistics. Therefore, we developed an OSS navigator: a Web-based dashboard for community-oriented reflection of OSS projects. The navigator was filled with data from communication and development repositories of three large bioinformatics OSS projects. The members of these OSS communities tested the prototype. The bioinformatics OSS developers acknowledged the uniqueness of statistics that the NOSE dashboard offers. Especially, graph visualization of the project social network received the highest attention. This network view combined with other community-oriented metrics can significantly enhance the existing visualizations or even be provided as a standalone tool.

1 Introduction

Success of an OSS project is tightly interwoven with the success of its community [Ray99], [HK03]. OSS systems co-evolve strongly with their communities [YNYK04]. Thus, the more successful a project is, the higher is the degree of its complexity in terms of project structure and community size. The complexity affects the awareness of community members of what is happening in their community. In interviews with OSS developers Gutwin et al. in [GPS04] find out that the awareness of other developers within OSS projects is essential for an intact project life. Within the study, project mailing lists (MLs) and text chats are determined as the main resources for maintaining group awareness. However, in large OSS projects, it gets very difficult for community members,

L. Corral et al. (Eds.): OSS 2014, IFIP AICT 427, pp. 11–20, 2014.

especially for the less experienced ones, to establish a complete and correct perceptional awareness model. In such cases, Gutwin et al. suggest to develop new representation methods for communication and its history.

Considering OSS mining research, there are already studies concentrating on OSS communication analysis: to investigate social network structure [BGD+06], to analyze content [BFHM11], to estimate the sentiment within OSS communities [JKK11]. OSS communication repositories reflect complete communities of the corresponding projects. In contrast, OSS source code repositories are restricted to the developers only. If we take a look at the OSS visualization platforms (e.g. GitHub, Ohloh, etc.), then they are focused either on source code or individual contributors. Platforms which provide OSS metrics based on project communication are still missing. To investigate this research niche, we address the following research question: **Are the OSS communities interested in a platform reflecting community evolution and if so, what evolution aspects should it reflect?**

The rest of the paper is organized as follows. Section 2 provides an overview on related systems for OSS project evolution visualization. To address our research questions, we executed an iterative study (Section 3). The achieved results are presented in Section 4. Section 5 concludes the paper and gives an overview of some ideas for future work.

2 Related Research

There are already plenty of related applications available for OSS development visualization. To give an overview of existing concepts and principles, the more notable ones are presented.

GitHub[1] offers a web-based hosting for software projects. Additionally, it provides visualizations focused mainly on project source code (commit activity, code amount) and some statistics on project contributors (contributor activity, followers and following people, projects, organizations, etc). Another popular web-platform for software projects' hosting is **SourceForge**[2]. It offers just some statistics on project traffic (hits on the project, number of downloads) and SVN activity. What statistics are visible to users depends on the project settings. In contrast, a web-service **Ohloh**[3] does not host the actual source code, but simply crawls and analyzes the OSS data. Ohloh offers many charts regarding the source code and contributors (their ranking and activity). Pure statistics are transformed into textual statements. Ohloh also provides data on project estimation effort based on the COCOMO model for software cost development [Boe81]. The next web front-end **Melquiades**[4] provides visualization for the data collected within **FLOSSmetrics**[5] research project [HIR+09]. The supported analysis is

[1] GitHub, `https://github.com`, last checked 2013/09/10

[2] SourceForge, `https://sourceforge.net`, last checked 2013/09/10

[3] Ohloh, `www.ohloh.net`, last checked 2013/09/10

[4] Melquiades, `melquiades.flossmetrics.org`, last checked 2013/09/10

[5] FLOSSmetrics, `flossmetrics.org`, last checked 2013/09/10

divided into three different types according to data resource used: data from source code repositories, data from mailing lists archives and data from tracker system repositories. However, not all projects have data regarding all three resources. Melquiades offers important metrics, like activity over time, growth and member inflow rate. The next two visualizations **Open Source Report Card (OSRC)**[6] and **Sargas** [SBCS09] provide contributor-oriented metrics. Based on the data from GitHub OSRC establishes developers profiles based on their daily and weekly activities, project participation, etc. Whereas, Sargas estimates the social profile of contributors based on their behavior within four social networks: open discussion forum, developers discussion list, discussions about the bugs and social network extracted from the source code.

To summarize, the existing applications are ranging from source code hosting services with visualization tools to pure analysis and visualization platforms. The last clearly proves the need and the interest of the OSS communities in self-monitoring tools. However, the existing systems focus mainly either on the system source code or on individual contributors. For monitoring of community evolution the information need to be presented from different perspectives.

3 Study Settings

Figure 1 represents the workflow of our study. To find out if the OSS members are interested in the community-related reflection of their projects, we first conducted an online survey within OpenStack, PostgreSQL, GIMP, Mozilla, Oracle VM VirtualBox, GNOME, TomCat OSS communities. The survey addressed questions related to the developer interest in a community-oriented metrics and what metrics are missing in the existing OSS navigators. Most of the questions had an optional comment field. We contacted OSS developers via the Internet Relay Chat (IRC) channels. The survey was anonymous, therefore, it was not possible to trace from which project the participants originated. Nevertheless, based on the survey results and by observing each chat for the next four hours, no malicious users were detected. The result of the survey was a positive answer to the first part of our research question. The OSS members do have strong interest in platforms reflecting OSS community evolution. Additionally, the OSS members suggested several ideas for metrics/aspects, which were assumed to be important to be aware of. To evaluate the feasibility of the collected ideas, we next applied prototype testing.

We selected a dashboard as a technological approach. *"Dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance"* [Few06]. The goal of the developed Navigator for OSS Evolution (NOSE) is to provide community-oriented navigation support for OSS project members. We filled the NOSE dashboard with the data from three long-term bioinformatics OSS (BioJava, Biopython and BioPerl), which have been already analyzed in our previous studies (e.g. [HK13]). Therefore, we

[6] Open Source Report Card, `osrc.dfm.io`, last checked 2013/09/10

were able to proceed with the prototype testing immediately after the development. An iterative development process of the NOSE dashboard was executed. Before starting the survey with bioinformatics communities, the dashboard was evaluated with 10 computer scientists. This evaluation was used to identify the design shortcomings of the developed dashboard.

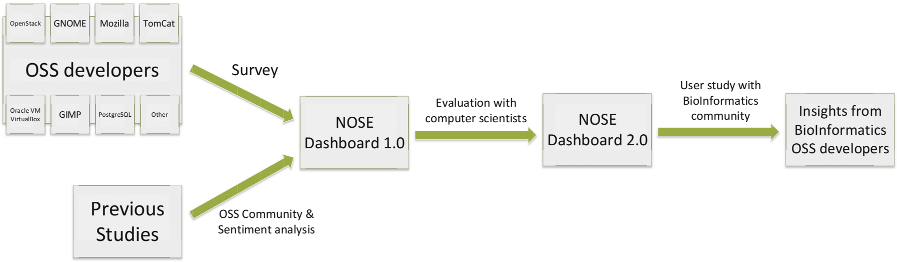

OpenStack
GNOME
Mozilla
TomCat
OSS developers
Oracle VM VirtualBox
GIMP
PostgreSQL
Other
Survey
Previous Studies
OSS Community & Sentiment analysis
NOSE Dashboard 1.0
Evaluation with computer scientists
NOSE Dashboard 2.0
User study with BioInformatics community
Insights from BioInformatics OSS developers

Fig. 1. Study Workflow

The relevance of the presented metrics and the data quality reflected in NOSE were directly investigated within bioinformatics OSS communities. We contacted the bioinformatics OSS community members via private emails. Such textual inquires encourage the participants to explain their answers and, thus, provides a more detailed feedback. Moreover, informal email exchange could trigger a fruitful discussion. The sent out email consisted of a short description of the NOSE goal and three questions:

- Do you find that the visualization features offered by GitHub are sufficiently informative?
- Would you additionally like to have features to represent the community and its structures?
- If so, would network graph be a viable option?

4 Navigator for OSS Community: Evaluation Results

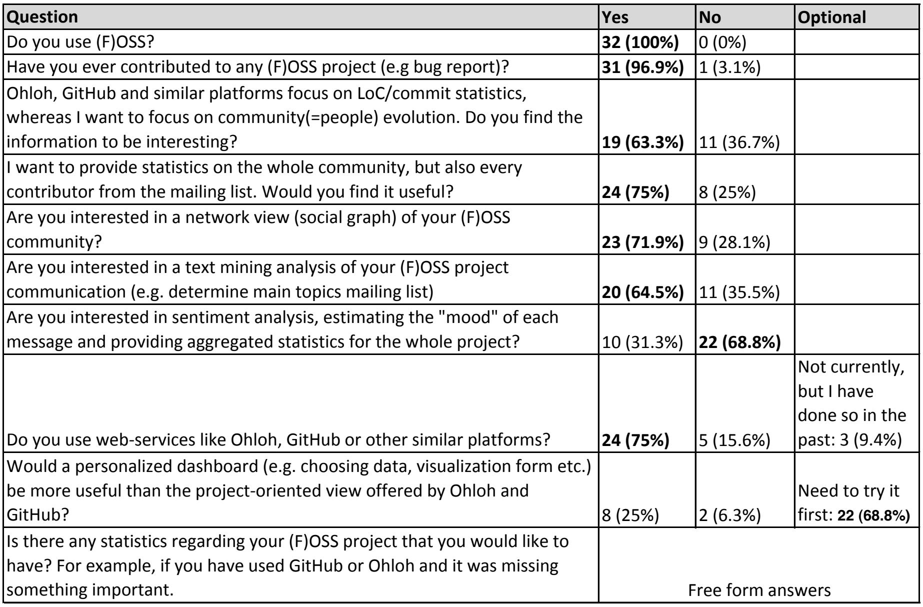
In following the results of both conducted surveys and prototype testing are presented.

4.1 Survey of OSS Developers

From OSS developers we received 32 responses with many (49) comments. Some developers provided initial feedback directly in the chat. Everyone who started the survey also finished it, what indicates the true interest of the participants in the survey. Figure 2 displays the survey questions with the collected feedback. Every question was optional, therefore some questions had less than 32 answers.

Web-platforms like Ohloh or GitHub were used by 75% of the participants. There were 16 comments in total, with GitHub being mentioned 13 times, SourceForge 5 times and Ohloh 4 times. 63.3% of the OSS developers were interested in the statistics related to community evolution. However, in four of the seven comments the participants mentioned, that it was unclear what kind of information was meant or *"How would/could I benefit from those information?"*

Social Analysis. The OSS developers were mostly interested in getting statistics from the MLs regarding the whole community. MLs were recognized as a useful source of information for getting an approximate user base size. However, one participant also mentioned a negative aspect, that *"[...] too much statistical evaluation could put the community of as they feel 'observed' "*. The most opinions of network graph representation of an OSS community were again positive with only two answerers mentioned that they would find it more interesting than useful.



Question	Yes	No	Optional
Do you use (F)OSS?	**32 (100%)**	0 (0%)	
Have you ever contributed to any (F)OSS project (e.g bug report)?	**31 (96.9%)**	1 (3.1%)	
Ohloh, GitHub and similar platforms focus on LoC/commit statistics, whereas I want to focus on community(=people) evolution. Do you find the information to be interesting?	**19 (63.3%)**	11 (36.7%)	
I want to provide statistics on the whole community, but also every contributor from the mailing list. Would you find it useful?	**24 (75%)**	8 (25%)	
Are you interested in a network view (social graph) of your (F)OSS community?	**23 (71.9%)**	9 (28.1%)	
Are you interested in a text mining analysis of your (F)OSS project communication (e.g. determine main topics mailing list)	**20 (64.5%)**	11 (35.5%)	
Are you interested in sentiment analysis, estimating the "mood" of each message and providing aggregated statistics for the whole project?	10 (31.3%)	**22 (68.8%)**	
Do you use web-services like Ohloh, GitHub or other similar platforms?	**24 (75%)**	5 (15.6%)	Not currently, but I have done so in the past: 3 (9.4%)
Would a personalized dashboard (e.g. choosing data, visualization form etc.) be more useful than the project-oriented view offered by Ohloh and GitHub?	8 (25%)	2 (6.3%)	Need to try it first: **22 (68.8%)**
Is there any statistics regarding your (F)OSS project that you would like to have? For example, if you have used GitHub or Ohloh and it was missing something important.	Free form answers		

Fig. 2. OSS Developers Survey Results

Text Mining (TM). Communication presents not only a source for social network analysis. It also provides a great unplugged pool for TM. TM methods allow to determine end-user requirements, discover conflicts, etc. Special area of TM is sentiment analysis. The mood of a user can be implicitly estimated based on opinionated documents generated by the user (e.g. postings in MLs). However, the OSS developers were mostly uninterested in sentiment analysis. The negative reaction could be the result of little awareness of the sentiment analysis meaning. For example, one of the participants said that *"Only slightly interested. I doubt that much useful information could be drawn from such an analysis, but I would need to know more about the methodology and findings to be sure, and it sounds interesting at least"*. Another participant expressed the concern that such analysis *"Would be interesting/fun, but i m not sure whether its useful for me [...]"*. However, there were also participants that were clearly in favor of sentiment analysis: *"Definitely. I would have stopped before entering some projects if I would have known about the mood swings of their contributors beforehand..."*. In contrast, other responses were clearly against it, for example *"I don't think public statistics of the form "messages from developer X are mostly aggressive" would do anything good"*. The concern of feeling observed was already mentioned in the context of social analysis. Consequently, the developers are rather interested in the aggregated statistics. For example: *"I think this [sentiment analysis] is only useful if combined with a certain segmentation of the user groups"*.

Majority of the participants were interested in TM analysis of the ML communication for other purposes:

- *"[...] determine the needs of the users in addition to voting and tagging in bugtrackers"*
- *"[...] creating FAQs for new contributors"*
- *"[...] finding out in which direction the community wants to evolve"*

Missing Functionalities. Finally, the following statistics were missed in the existing OSS visualizations:

- *"[...] which wiki pages are consulted often, which problem appear in lists/ forums frequently and so on."*
- *"Some projects [...] allow Users to be credited in commits for there Testing or Bug reports (e.g. Reported-By: Tested-By:) including those contributions in statistics would by nice [...]"*
- *"Last time on ohloh I wanted to see a simple list of contributions, but I only found timelines and such, which I find hard to browse"*
- *"More informations about project activity. Most FOSS project have stalled development, are abandoned. This is for me the #1 FOSS problem"*

Dashboard. A personalized dashboard was considered useful by 25% of the participants, while the majority of the participants (68.8%) replied that they would need to try it first. One participant mentioned that *"The projects I contribute to have their own dashboards, I don't think an external dashboard would match my expectations"*. Indeed, many of the OSS hosting platforms offer their own built-in analysis and visualization.

Summarized, the OSS developers showed a strong interest in both social and text-based community communication analysis. To find out which statistical charts and designs were truly useful, there had to be an application that the OSS developers could try out.

4.2 Bioinformatics OSS Developers

Figure 3 displays a screenshot of the NOSE dashboard evaluated by bioinformatics OSS developers. It consists of five widgets: inflow vs. outflow of members in project MLs, number of commits, sentiment within community vs. commit activity, size of community core, social network graph of the project community. The last widget additionally provides several options: to search for a person, to select a yea or a release, and to highlight the core.

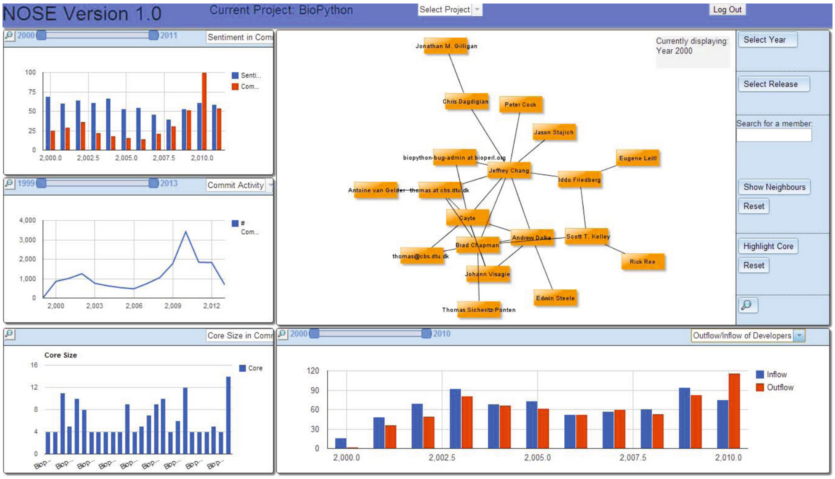


Fig. 3. Screenshot of the NOSE Dashboard Prototype

The survey was sent to members of all three bioinformatics communities. We selected the participants who were active in the MLs in the last two years. In total, we sent an email to 46 project participants. From the 46 persons, nine replied to our email. The participants were also open for the discussion, thus, many issues got an extensive clarification. One of the BioPython developers even posted an article about the NOSE dashboard to his blog[7], which again supports the community interest in the self-monitoring and -reflection.

The proposed community metrics mainly received a positive feedback from the bioinformatics OSS developers. However, three bioinformatics developers reported that, they were not using the OSS visualizations. Nevertheless, one of them mentioned that the NOSE dashboard looked fairly attractive, but he was unsure what he would use it for. Another developer gave a longer explanation saying that *"I don't find this type of community information particularly useful. Over time I've developed a habit of doing my own, informal, reputation scores*

[7] 'The Bio* projects: a history in graphs", `http://bytesizebio.net/2013/09/07/bio-projects-a-history-in-graphs`, last checked 2013/11/27

in my head, based on people's list participation and tone, their code, their constructive criticisms, etc.... Who's talking to whom, etc... has never been particularly useful". Similar skepticism among OSS developers was previously reported in [GPS04]. Despite some critical opinions, other evaluation participants were more in favor of the community statistics. One BioJava developer stated, that although BioJava project currently uses Ohloh for visualization, the NOSE dashboard could be complementary to existing visualization platforms. The developer added that these kinds of statistics would be useful for recruiting and funding. *"For instance, we use these types of stats when applying for Google Summer of Code sponsorship"*.

The network graph received by far the most praise and interest. One developer commented that adding *"[...] the social graph to the existing GitHub facilities would be valuable"*. Another reply was: *"The social aspects of OSS projects are no less intriguing than the technological ones!"*. The developer additionally named two gains from such statistics. Firstly, that visualization platforms like the NOSE dashboard are great for getting an overview of the project's history. Secondly, that *"[...] there is a lot to learn from this on how OSS projects get off the ground, what makes a successful project, etc"*.

4.3 Discovered Weaknesses

More Data Sources. Many of the developers mentioned that additionally it would be nice to have data from GitHub. One developer expressed interest in comparing the social networks created based on GitHub and ML. GitHub is *"a great place to discuss code specifics, so is often easier to go back and forth on than writing e-mails. It would be cool to see how the interactions there overlay on this"*. Another developer expressed interest in getting such communication statistics from the project LinkedIn[8] group.

Network Graph. Broadcasts were excluded from the network graph visualization. If the broadcasts were included, it would create an enormous amount of edges. That would make the rendering of a comprehensible network almost impossible. Additionally, it would create many hubs, thus lowering the presence of actual core developers. Further, one of the evaluation participants identified one core developer, who was split into two aliases. This splitting decreased his/her social role in the network graph. Nevertheless, bioinformatics developers believed the network graph captures the community quite accurately.

There were many suggestions and requests. Most of the feature requests were directed at getting statistics from additional sources and not only from the ML. Some metrics requests were related to the social network graph:

– *"[...] add a graph to the dashboard that shows, for each year who are the top linked nodes"*

[8] `http://www.linkedin.com/groups/BioJava-58404?home=&gid=58404&trk=anet_ug_hm`, last checked 12.09.2013

- *"[...] graph comparative metrics, such as consensus linkage, slope of the edge-number histogram [...]". The developer suggested that it could be used for comparing the three bioinformatics projects.*
- *"[...] use the graph's connecting edges to indicate the strength/weight of the the connection (i.e. line thickness linked to number of email conversations)"*

5 Conclusions and Future Work

In this paper, we addressed the research question: Are the OSS communities interested in a visualization platform, which reflects community evolution? If so, what aspects should it reflect? To answer this research question, we surveyed members from different OSS communities. Based on the survey results, we developed a dashboard prototype for community-oriented navigation in OSS projects. The evaluation within three long-term bioinformatics OSS showed a strong interest of OSS developers in visualization of community statistics. Especially, the network graph visualization of the communities was recognized as the most interesting metric. The developers are more interested in aggregated statistics in order to avoid the feeling of being observed among the project participants. On contrary, sentiment analysis did not get much attention, which might be a result of a poor description or little awareness of the analysis method. However, some evaluation participants saw the NOSE platform more as fun, than as a useful evolution barometer. Further, the dashboard was suggested as a possible extension for the existing platforms and not as a standalone application.

Our next steps are to realize the identified requirements. In terms of analysis metrics, the OSS members wish topic-based text mining [GDKJ13] measures, with the goal to see where users struggle. Considering the data, there are many requests to extend the data sources, for example by the data from GitHub. Further studies with domains outside bioinformatics are needed to achieve truly generalizable results. Currently, we apply the concept of the NOSE dashboard to support and manage an OSS community around a EU project Learning Layers[9].

Acknowledgement. This work is supported by the LAYERS FP7 ICT Integrated Project (grant agreement no. 318209) of the European Commission.

References

[BFHM11] Bohn, A., Feinerer, I., Hornik, K., Mair, P.: Content-based social network analysis of mailing lists. The R Journal 3(1), 11–18 (2011)

[BGD$^+$06] Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of the 2006 International Workshop on Mining Software Repositories, MSR 2006, pp. 137–143. ACM, New York (2006)

[Boe81] Boehm, B.W.: An experiment in small-scale application software engineering. IEEE Transactions on Software Engineering 7(5), 482–493 (1981)

[9] Learning Layers, `http://learning-layers.eu`, last checked 2013/11/10

[Few06] Few, S.: Information Dashboard Design: The Effective Visual Communication of Data, p. 35. O'Reilly Media (2006)

[GDKJ13] Günnemann, N., Derntl, M., Klamma, R., Jarke, M.: An interactive system for visual analytics of dynamic topic models. Datenbank-Spektrum 13(3), 213–223 (2013)

[GPS04] Gutwin, C., Penner, R., Schneider, K.: Group awareness in distributed software development. In: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, CSCW 2004, pp. 72–81. ACM, New York (2004)

[HIR$^+$09] Herraiz, I., Izquierdo-Cortazar, D., Rivas-Hernandez, F., Gonzalez-Barahona, J., Robles, G., Duenas-Dominguez, S., Garcia-Campos, C., Gato, J.F., Tovar, L.: Flossmetrics: Free/libre/open source software metrics. In: 13th European Conference on Software Maintenance and Reengineering, CSMR 2009, pp. 281–284 (2009)

[HK03] von Hippel, E., von Krogh, G.: Open source software and the "private-collective" innovation model: Issues for organization science. Journal on Organization Science 14(2), 208–223 (2003)

[HK13] Hannemann, A., Klamma, R.: Community dynamics in open source software projects: Aging and social reshaping. In: Petrinja, E., Succi, G., El Ioini, N., Sillitti, A. (eds.) OSS 2013. IFIP AICT, vol. 404, pp. 80–96. Springer, Heidelberg (2013)

[JKK11] Jensen, C., King, S., Kuechler, V.: Joining free/open source software communities: An analysis of newbies' first interactions on project mailing lists. In: Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS), pp. 1–10 (January 2011)

[Ray99] Raymond, E.S.: The Cathedral and the Bazaar. O'Reilly Media (1999)

[SBCS09] de Sousa, S.F., Balieiro, M.A., dos R. Costa, J.M., de Souza, C.R.B.: Multiple social networks analysis of floss projects using sargas. In: 42nd Hawaii International Conference on System Sciences, HICSS 2009, pp. 1–10 (2009)

[YNYK04] Ye, Y., Nakakoji, K., Yamamoto, Y., Kishida, K.: The co-evolution of systems and communities in free and open source software development. In: Koch, S. (ed.) Free/Open Source Software Development, pp. 59–82. Idea Group Publishing, Hershey (2004)