Open Source Resume (OSR): A Visualization Tool for Presenting OSS Biographies of Developers

Thunyathon Jaruchotrattanasakul*, Xin Yang†, Erina Makihara†, Kenji Fujiwara†, Hajimu Iida†
*Kasetsart University, Thailand
thunyathon.j@ku.th
†Nara Institute of Science and Technology, Japan
{kin-y, makihara.erina.lx0, kenji-f}@is.naist.jp, iida@itc.naist.jp

***Abstract*—In order to recruit appropriate developers for software projects, it is important to have a clarified understanding on the practical experience and expertise of each candidate. However, traditional resume only shows experiences claimed by developers, and very few evidence or information regarding their actual development activities can be obtained. In this paper, we propose an approach to extract developers' practical activities from their participated open source software (OSS) projects, and generate the biographies that reflect their OSS contributions. We applied the approach on the largest code hosting service, GitHub. By investigating the resumes generated from the extracted dataset, recruiters of software projects can be given a clearer view on whether the developers' experiences fulfill the qualifications. Moreover, based on the approach, we present a web-based visualization tool, named as Open Source Resume (OSR). We believe our tool is useful to help recruiters from software development organizations to search for suitable developers, and then construct development teams in software projects based on their OSS contributions.**

***Index Terms*—Open Source Software; Mining Software Repositories; Developer Expertise; Visualization Tool**

I. Introduction

Open Source Software (OSS) has been growing rapidly since last few decades. It transformed from individual project developed under the volunteer-based collaboration, to an evolving common platform that even commercial software organization participates as another stakeholder for development. [3]. Many organizations are not only involved in the development of OSS products, but they also recruit the corresponding OSS developers with remarkable skills [5] [10].

However, during the progress of recruiting software developers, traditional resumes only show experiences claimed by developers, and very few evidences or information regarding their actual development activities can be obtained. Candidates may cheat on their resumes by fake programming skill or project experience. As recruiters, it is difficult to examine the truth of the candidates' OSS development activities according merely to their resumes. For addressing this issue, given the high transparency of OSS, rich data can be utilized for helping software development organizations to make a more accurate decision on hiring suitable developers.

In this paper, we propose an approach to extract developers' practical activities from their OSS contributions. The extracted data provides affirmative information that reflects their actual project experiences and expertise. Furthermore, we developed a tool, named as Open Source Resume (OSR), to visualize the data for presenting the OSS biographies of developers. As our preliminary work, we applied OSR on the GitHub community[1]. GitHub, is not only the largest code repository hosting service, but also a large-scale social community for developers. GitHub users are able to collaborate, increase their expertise and establish their reputation in the community [2]. The source code of our tool is available to access[2].

The following two features are visualized in OSR for assisting software development team to recruit OSS developers:

(1) Querying for suitable candidate OSS developers. (2) Presenting detailed OSS biographies of developers based on their OSS contributions. With these two major features, main contributions of this paper can be summarized as follows:

- We proposed an approach to construct and visualize developers' OSS biographies based on their contribution activities.
- Based on the approach, a visualization tool was implemented with another main feature of querying for OSS developers based on user's criteria.
- The tool is considered to provide a convenient way to seek suitable developers for constructing development teams in software development.

The remainder of the paper is organized as follows: Section II describes the existing related work. Section III introduces the two main features of OSR and the respective motivation and visualization for each feature. Section IV presents the technical details about the implementation of OSR, while Section V discusses the limitations that exist in the current stage of implementation. Finally, Section VI concludes the whole paper and presents the future work.

II. Related work

In this section, we introduce prior work related to this paper. Social Network Service(SNS) has been one of the most popular topics around the world since the last decade [1]. As a result of the rapid increase in SNS users, some studies applied data mining techniques to analyze SNS for exploring trending topics and seeking desirable community for oneself [7].

[1]https://github.com/
[2]https://github.com/lostseaway/OSR

978-1-5090-1851-2/16 $31.00 © 2016 IEEE
DOI 10.1109/IWESEP.2016.17

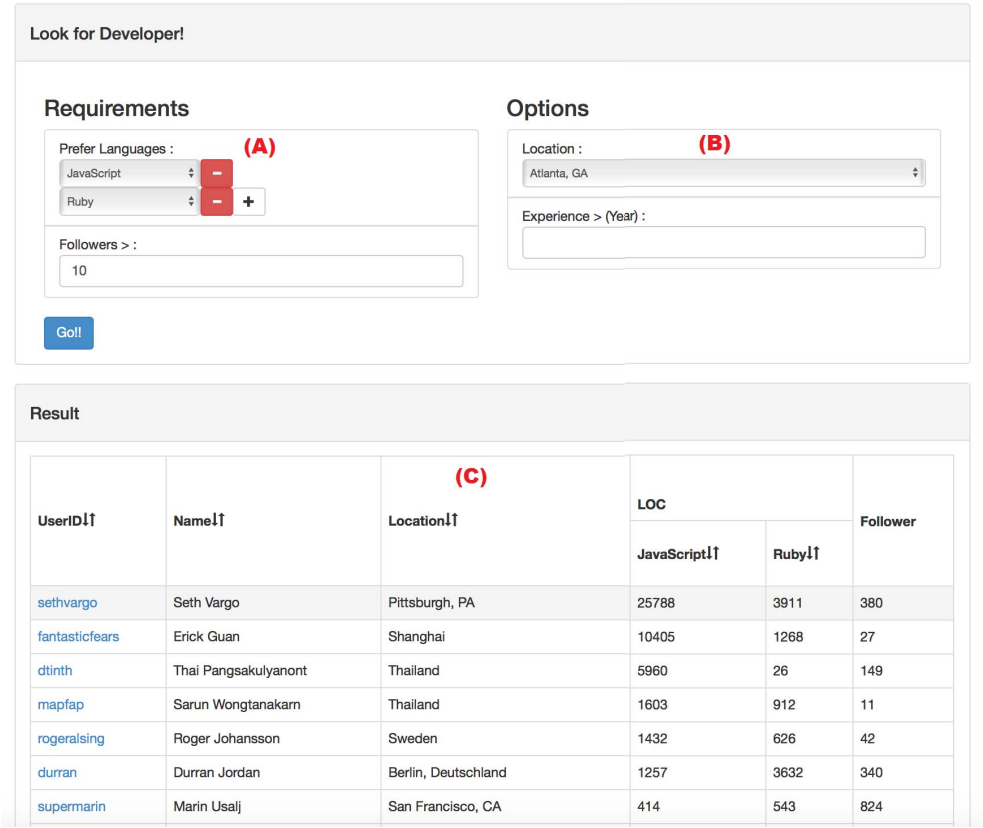

Figure 1. An example of querying GitHub developers and presenting the results list using OSR

GitHub, as the largest OSS developers' community, is also explored and analyzed for the purposes of visualizing OSS community or identifying OSS contributors. In recent year, many organizations pay remarkable attention to GitHub data analysis [1]. Gousios et al. collected significantly large amount of data from GitHub by using official API and such data was proceeded to form a project, named as GHTorrent, to present the dataset of public projects in GitHub [4]. Hauff et al. proposed the pipeline for automatically matching between job advertisements and developers on GitHub [5]. However, they collected developer's profile through README, in which the concerned data for our study is not necessarily included. (e.g., the region developer lives or year of programming experience). Therefore, we focus on using GitHub API to collect profile data.

Some studies especially focus on developers' profile data in GitHub. Shami et al. described how people use the information gained from viewing online profiles to determine the most suitable candidate to contact for help on a topic [8]. They pointed out that public profile information influences impressions of work-related skills. Marlow et al. state that GitHub consists of many unknown developers, so developer profile design should be optimized for efficiently visualizing or summarizing the information for quick perusal [6]. Singer et al. propose profile aggregators to evaluate developers based on their social development platforms [9]. As mentioned in previous studies, the profile on GitHub is an important part for users and contributors to seek for desirable information.

III. Overview of OSR

OSR is a web-based application to help software development teams to recruit developers based on their OSS contribution activities. This tool provides insights into developers' practical contributions to OSS projects, which can be an useful criteria for judging whether they are suitable for joining the development team based on their different qualifications. In this section, we explain the two main features of OSR, (1) querying for developers, and (2) presenting developers' OSS biographies. Moreover, for each feature, we explain the respective motivation and how the visualization is planned to be achieved. In this paper, by utilizing the data source retrieved from GitHub code repository, we construct OSR based on developers' OSS contributions in GitHub.

A. Query for OSS Developers

Motivation. Choosing the most suitable developer from available candidates is an important task during the recruitment for software development. Moreover, the recruitment requirements vary in accordance with the objectives of projects.

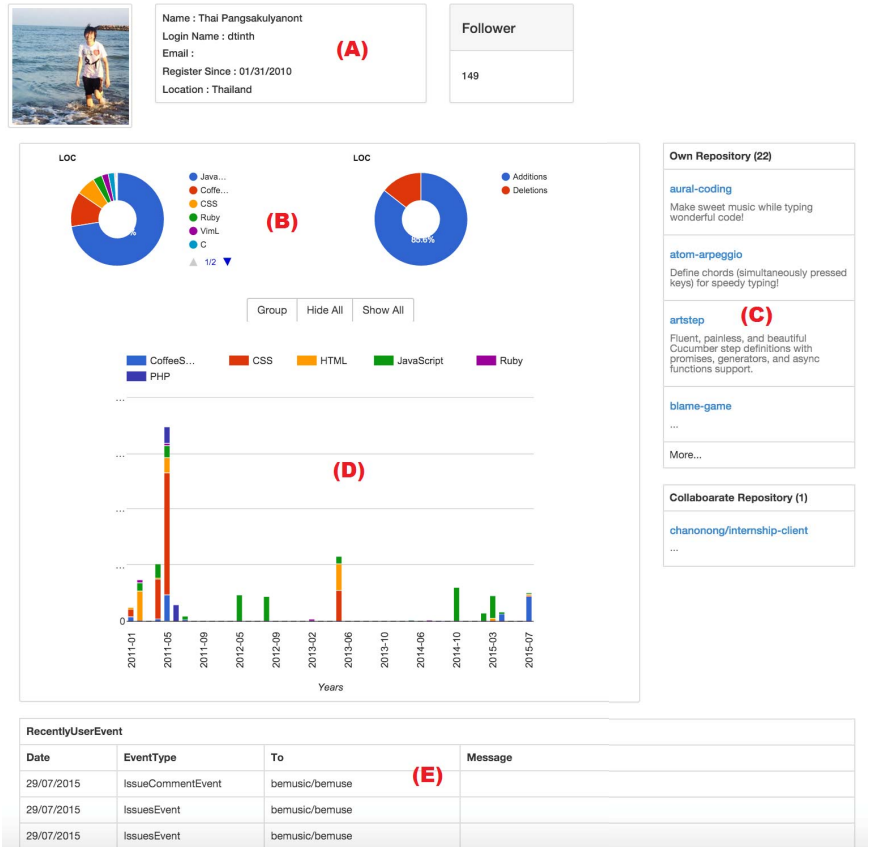

Figure 2. An example of presenting an OSS biography of a GitHub user using OSR.

Considering that massive number of developers are involved in OSS contributions through GitHub, the repository hosting service serves as a resourceful and trustful platform for recruiting demanded developers with remarkable skills. However, there exists no efficient tool to query for suitable candidate developers based on their OSS-related activities. Therefore, we create our tool based on: (1) Experience and expertise of potential candidates: For example, preferred programming language, year of experience using these languages, etc. (2) Location of developers: Although OSS development can be achieved by collaboration between geographically distributed developers, location is still critical in recruitment. For example, commercial organization may recruit some local developers who can work as full-time employees. (3) Social relationship: Some developers are more eager to work alone on their own repositories, while some developers are likely to collaborate with people and establish their reputation among the community.

Visualization. Figure 1 is a practical example of querying for GitHub developers and presenting the result list in OSR. First and foremost, the user (recruiters) can select and input their criteria for searching appropriate OSS developers. Multiple criteria options are provided for users to choose from *Options* list on the right side of query box, as shown in Figure 1(B). For creating the search criteria, the required options can be drag-and-dropped to the *Requirement* box on the left side as shown in Figure 1(A). Once the preference information is input and the *Go!!* button is pressed, a query is sent to search through the cache database, which is composed of developers' data retrieved from GitHub. As a result, a list of OSS developers that match the user's criteria is displayed underneath the query box, as shown in Figure 1(C). The result list shows the criteria options as column name, and users can sort the results by different columns according to their needs. Furthermore, an alternative way of search by GitHub user's id is also provided in the tool. As an usage example in Figure 1, we queried for the developers who are good at both JavaScript and Ruby, and the developers should be rather popular with more than ten followers in GitHub. After pressing the query button, we obtained a result list consisting of candidates who match with the input criteria. Since we were more interested in their JavaScript experiences (represented as LoC in the list), we sorted the results by JavaScript column as well.

B. Developers Contribution Visualization

Motivation. Traditional resumes, either in hard-copy or electronic style, only show the experiences claimed by candidates, and very few pieces of evidence regarding their actual

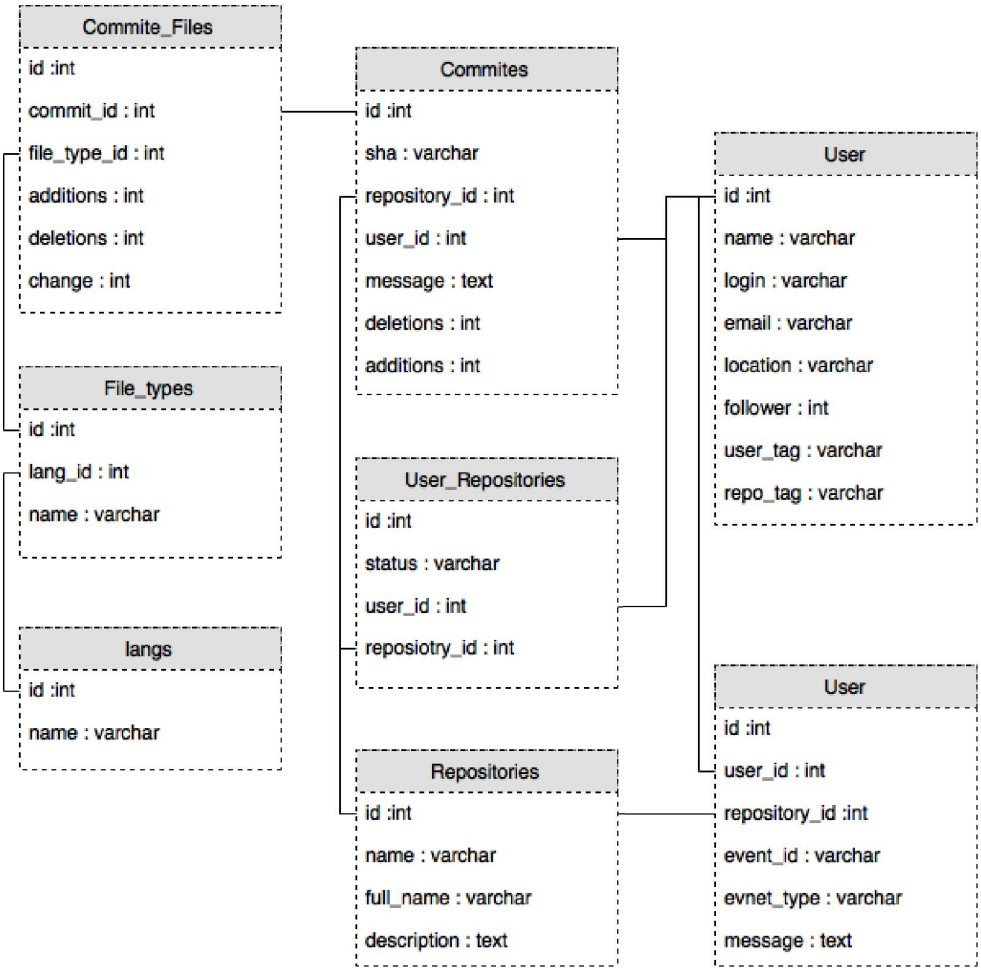

Figure 3. The relational database schema.

development activities can be obtained. However, for those who claim to have OSS projects experiences, such experiences can be evaluated by extracting their actual contributions from those claimed projects. Moreover, the extracted data can be used for automatically generating practical OSS biographies of developers, to which Project leader or human resource department can refer for making a more accurate judgment, and simultaneously recruiting potentially remarkable developers.

Furthermore, visualization of such OSS biography is considered to provide even better insight into the practical experience, expertise, behaviors, and social relationship of candidate developer, comparing to traditional resumes.

Visualization. We visualize this feature by building a developer's profile page as shown in Figure 2. The main objective of this page is to demonstrate a clear overview of OSS contributions as candidate's contribution biography. This OSS biography is composed of following five main sections:

1) Developer's personal information as Figure 2(A).
2) Analyzed coding behaviors based on their committed source code as Figure 2(B).
3) List of associated public repositories as Figure 2(C).
4) Timeline of contribution activities as Figure 2(D).
5) Recent activities list as Figure 2(E).

Figure 2(A) shows several basic personal information such as the Name, GitHub id, Email address, and others. The number of followers indicates the "popularity" of the developer. As the most significant feature, Figure 2(B)(D) displays visualized data of developer's coding activities that correspond to OSS contributions. As shown in Figure 2(B), the left pie chart indicates the contributed source code written in respective programming languages. In addition, the one on right side refers to overall code modification activities including additions and deletions on all associated public repositories. Furthermore, the stacked bar chart as shown in Figure 2(D) shows coding activities of different programming languages in a chronological order. We believe all these statistics charts provide a clear overview of this developers' OSS contribution activities. As shown in Figure 2(C), the developer's associated public repositories, which include self-owned repositories and other repositories that developer makes contributions to,

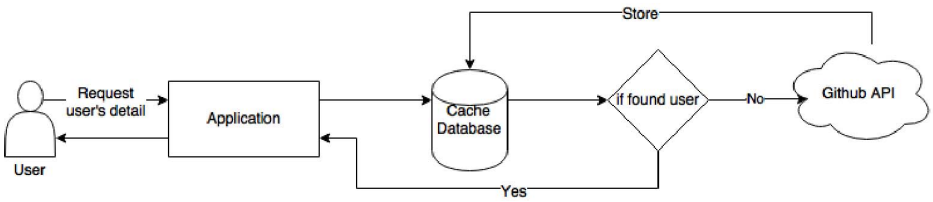

Figure 4. The workflow of searching developers in OSR.

Table I
A LIST OF GITHUB API URL.

URL Path	Description
/users/<login ID>	GitHub's user infomation.
/users/<login ID>/repos?type=all	Get all public repository of the user.
/repos/<repo's name>	Get repository's detail.
/repos/<repo's name>/languages	Get LOC of each Prog. Language in repo.
/repos/<repo's name>/commits?author=<login ID>	Get all Commit of the user from the repo.
/user/<login ID>/events	Get all user's event.

Table II
DATABASE TABLE DESCRIPTION

Table Name	Description
users	GitHub's user infomation.
user_repositories	Repositories of GitHub's user.
repositories	Repository's infomation.
lang_repositories	Programming Language of the repository.
lang	Programming Laguage's infomation
commits	Infomation of user's commit.
commit_files	File's infomation in the commit.
file_types	Type (Programming Language) of file.

is listed. These repositories histories indicate corresponding information about the project details and working behaviors of the developer. From the list, recruiter can obtain the information of developer's associated projects, and how the developer participating in these projects. At last, as shown in Figure 2(E), recent OSS-related activities are demonstrated as well.

IV. IMPLEMENTATION

For implementing OSR, two major steps are involved: Constructing data source from GitHub raw data, and Designing an interactive tool that also visualizes necessary data.

A. Constructing Data Source

For composing persuasive developers' profile through OSS activities, multiple information such as developers' personal information, their public GitHub repositories, their used programming languages for making contributions is necessary. The respective raw dataset of such information, which is represented in JSON format, can be obtained from GitHub through the GitHub REST API, as shown in Table I. For better efficiency, multiple worker threads are created for sending different API requests and retrieving the corresponding dataset simultaneously. The dataset is then parsed and stored at a locally created MySQL-based cache database. We provide the data in a relational database structure as shown in Figure 3 for convenience. Moreover, we provide the descriptions of the tables in the database as shown in Table II.

The main purpose of this cache database is performance-wise, which reduces redundant API requests to GitHub. As indicated in Figure 4, whenever a user requests for developer's details by searching GitHub developer's id, rather than acquiring data by sending GitHub API, the data is searched through the cache database. If matched data is found, it will be presented to the user. Otherwise, a new API request is sent to GitHub for retrieving the demanded data, which is then appended to the cache database. As such, the amount of data within the cache database is self-expanding alongside with increasing requests from users. Furthermore, considering that developers' data is progressively changing, Etag is utilized to update existing data stored in cache database. Etag is a light-weight mechanism provided by HTTP, which allows the user to make a simple request for the verifying version of the data source. Therefore, triggered by every user's request, an API is sent to GitHub for acquiring the Etag of the demanded developer's information. If the received Etag is same as the one being stored in cache database, it means the data of the corresponding developer is already up-to-date. Otherwise, the data is updated through new API requests.

B. Tool Support

The web-based structure of OSR is implemented upon Ruby on Rails MVC framework. The main visualized features of the tool, query for OSS developers and presenting OSS biography, represent the two most significant *View* classes. Whenever a user creates a search request. The *Controller* class sends a corresponding MySQL query to the cache database. As the *Model* class of overall structure, to search for the matched OSS developers together with the associated data. The retrieved results are appropriately arranged for OSR user's

clear comprehension of data in the view pages. Particularly, for the pie charts and timeline that correspond to OSS contribution activities in OSS biography view page, such visualization is achieved by integrating D3.js, which is a powerful javascript library for visualizing data, into the framework.

V. Discussion

At the early stage of implementation, instead of retrieving data in real-time from Github by using the provided REST API, we took a small dataset available from GHTorrent to build up the cache database. Mainly for the purpose of experiment, the dataset contains information of 1000 selected Github users (OSS developers). These 1000 developers are composed of top 200 most experienced individuals each from 5 popular programming languages (e.g. Ruby, Python, C, C++, etc.), based on their code contributions in the respective category. However, as we attempted to expand the dataset by sending API to Github for acquiring new OSS developer's data directly, it takes around 10 seconds to process each request, which is considered to be not efficient. We plan to improve the performance of this particular workflow as part of the future works.

From here we discuss about several limitations that exist in the current stage of implementation. First, only data of Github users are available on OSR, as we will expand the availability to other repository hosting platforms and OSS projects. Furthermore, geographical location of OSS developers is not always presented because such information is seldom provided in their Github profiles. Moreover, OSR is not able to access private repository, which can become a potential threat to validation of the OSS biography.

VI. Conclusion

In this paper, we propose an approach to retrieve the practical activities of OSS developers from their contributions in OSS projects. Moreover, we implemented a visualization tool, named as OSR, to visualize the data for presenting the OSS biographies of developers. We applied OSR on the GitHub community as our preliminary work, and the tool is able to extract developers' profile data based on their coding experience, expertise, behaviors and social relationship. We believe our tool is able to help recruiters from software development organizations to search suitable and construct development teams.

As for our future plan, we will continue the development of OSR, and try to apply it on other repository hosting platforms and OSS projects (e.g., Apache, Eclipse, OpenStack, etc.). Moreover, we plan to add more search options and introduce a comparison feature into our system, to increase the accuracy and the efficiency when querying for developers. Finally, we hope to evaluate the usefulness of OSR by applying it to practical recruiting of software development teams. We believe that identifying and presenting the OSS contributions of developers are not only for development team recruiting, but also bring benefits to researches regarding human factor and social aspect in software engineering.

References

[1] M. Adedoyin-Olowe, M. M. Gaber, and F. T. Stahl. A survey of data mining techniques for social media analysis. *CoRR*, abs/1312.4617, 2013. II

[2] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1277–1286. ACM, 2012. I

[3] B. Fitzgerald. The transformation of open source software. *Mis Quarterly*, pages 587–598, 2006. I

[4] G. Gousios. The GHTorent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 233–236, 2013. II

[5] C. Hauff and G. Gousios. Matching GitHub developer profiles to job advertisements. In *Proceedings of the 12th Working Conference on Mining Software Repositories*, pages 362–366, 2015. I, II

[6] J. Marlow, L. Dabbish, and J. Herbsleb. Impression formation in online peer production: activity traces and personal profiles in GitHub. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 117–128. ACM, 2013. II

[7] M. A. Russell. *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. " O'Reilly Media, Inc.", 2013. II

[8] N. S. Shami, K. Ehrlich, G. Gay, and J. T. Hancock. Making sense of strangers' expertise from signals in digital artifacts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 69–78. ACM, 2009. II

[9] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider. Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pages 103–116, 2013. II

[10] R. T. Watson, M.-C. Boudreau, P. T. York, M. E. Greiner, and D. Wynn Jr. The business of open source. *Communications of the ACM*, 51(4):41–46, 2008. I