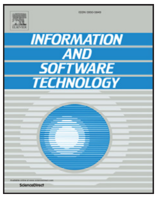
Information and Software Technology 70 (2016) 30–39

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof



# Understanding the popular users: Following, affiliation influence and leadership on GitHub

Kelly Blincoe[a,*], Jyoti Sheoran[b], Sean Goggins[c], Eva Petakovic[c], Daniela Damian[b]

[a] Auckland University of Technology, New Zealand
[b] University of Victoria, Canada
[c] University of Missouri, USA

ARTICLE INFO

*Article history:*
Received 12 April 2015
Revised 10 October 2015
Accepted 12 October 2015
Available online 20 October 2015

*Keywords:*
GitHub
Open source software
Social computing
Following
Influence
Leadership

ABSTRACT

*Context:* the ability to follow other users and projects on GitHub has introduced a new layer of open source software development participants who observe but do not contribute to projects. It has not been fully explored how following others influences the actions of GitHub users. *Objective:* this paper studies the motivation behind following (or not following) others and the influence of popular users on their followers. *Method:* a mixed methods research approach was used including a survey of 800 GitHub users to uncover the reasons for following on GitHub and a complementary quantitative analysis of the activity of GitHub users to examine influence. Our quantitative analysis studied 199 popular (most followed) users and their followers. *Results:* we found that popular users do influence their followers by guiding them to new projects. As a user's popularity increases, so does their rate of influence, yet the same is not true for a popular user's rate of contribution. *Conclusions:* these results indicate that a new type of leadership is emerging through GitHub's following feature and popularity can be more important than contribution in influencing others. We discuss implications of popularity and influence and their impact on social structure and leadership on OSS projects.

## 1. Introduction

Understanding how influence is exerted on social computing platforms is critical for participants and leaders because it impacts their patterns of work, interactions, and knowledge management in collaborative environments. While much work has been done to understand influence on purely social platforms like Twitter and Facebook [1–3], influence on social software development environments like GitHub has been understudied. GitHub enables users to "follow" individuals, much like one follows another user on Twitter. However, users on Twitter broadcast their own messages and diffuse only ephemeral information, links and perspectives 140 characters at a time. In contrast, notifications sent to GitHub followers embody evidence of work that has occurred and are not broadcast by the user, but rather, by the system [4]. The GitHub following feature is, therefore, interesting to study since structure and influence in OSS seem connected to the types of activities participants engage in.

Previous studies of influence in Open Source Software (OSS) found a "pyramid meritocracy" [5–7], where there is hierarchy and centralization, but usually not an authoritarian "great leader" at the top. Leadership is shared by a group of developers who "act as [influential] peers at the top echelon of the pyramid" [7]. The metaphor of a pyramid depicts new contributors at the bottom, "leaders" in the middle and "elders", who may have previously led projects but now advise, at the top. Elders are particularly common on long running OSS projects. OSS projects that are led by a software company may also have a community manager who acts as a liaison between the company and the OSS community [8]. With the advent of GitHub, new types of leadership are emerging because GitHub fosters a lower barrier to entry than other OSS platforms like SourceForge [9].

Other studies of OSS projects (e.g. [10–14]) reveal patterns of organization participants as "a series of concentric circles", where at the center are the core developers. The core developers are surrounded by a ring of "maintainers" who are responsible for one or more modules of a project. The outer-most circle contains the "patchers" who fix bugs, "bug reporters", "documenters" and "users". As Ducheneaut [10] points out, even users can be "highly skilled", suggesting that the "periphery" is a "nebulous arrangement of both skilled and unskilled individuals."

This nebulous skill arrangement is further extended on GitHub through its following feature, which allows users to "observe" without participating. This type of participation has the capacity to alter OSS engagement models. Our research goal is to understand how the lightweight connection of following another user builds into cascades

[*] Corresponding author. Tel.: +64 226107330.
*E-mail address:* kblincoe@acm.org (K. Blincoe).

http://dx.doi.org/10.1016/j.infsof.2015.10.002
0950-5849/© 2015 Elsevier B.V. All rights reserved.

of influence and to study the relationship between emerging social structures in projects on GitHub, the most followed (popular) users, and leadership in these projects. In this paper we take a first step in this direction by investigating the most followed users and their influence on their followers.

In this paper, we investigate following behavior and the influence of following others using a mixed-method research approach. We surveyed 800 GitHub users to identify motivations behind following other users on GitHub and why some users choose not to follow others. To examine influence of following, we conducted an analysis of the 199 most popular GitHub users (measured by number of followers). By analyzing the actions of popular GitHub users and their followers, we found that popular users often attract their followers to new projects. This is in line with previous results [15] that found that users who are both very popular and very active influence their followers. In this previous work, only users who fall into both of these categories were included; therefore, the two dimensions were intertwined and unable to be studied individually [15]. Instead, we study all popular users regardless of activity level. This allows a deeper investigation into the effect of popularity. Through our investigation, we found that the rate of a popular users' contribution does not impact her rate of influence. However, the rate of influence of a popular user does increase as the user accumulates more followers. Our findings indicate that GitHub's following feature may be enabling a new type of leadership in GitHub-hosted OSS projects and that popularity may be more influential than actual contribution.

### 1.1. GitHub: open, collaborative software development environment

GitHub is a web-based, social software development environment that provides source code management, issue tracking and other features. GitHub allows users to set up a public repository that anyone can fork and use for their own code and/or to contribute changes to the code. Pull requests are a way in which code from one developer is contributed back to a GitHub repository publicly. A "fork" is a clone or copy of a repository. Forks are made for two main reasons: first, to use the code in some derivative way; and second, as a precursor to contributing back to the original project through a pull request, which can then be merged with the main branch of code by those with access. All of these activities are published in an open, visible stream on GitHub, and content and discussion associated with issues, commits and pull requests are also public. Users of GitHub can receive alerts (via desktop clients and email) about code changes, pull requests, comments, issues, etc. for any public project.

Some of the social features on GitHub include following other users and starring other projects. Starring a project is a way for users to indicate that the project is interesting to them. Since there are often multiple GitHub OSS projects providing the same functionality, stars are often used as a 'vote' to indicate that the project is worth using and, thus, are a signal of the health of GitHub-hosted OSS projects.

## 2. Related work

Although research on ways of working in GitHub has been growing rapidly, aspects of project organization, social structures, and leadership and its influence on work performance in such open environments have not received much attention. Dabbish et al. [9] do provide insight into how GitHub's design enables new types of collaboration and collaboration patterns through increased platform transparency for individual users. They discuss the "collaborative utility" and value when "transparency is integrated into a web-based workspace" as well as the value of social features that make activity visible to users – GitHub supports learning "better ways to code and access to superior knowledge" [9]. Not only are GitHub users able to evaluate contributions by examining discussions around those contributions [16], GitHub's social interface also allows people to make inferences about other contributors' "technical goals and vision when they edit code" [9]. This transparency of activity allows users to define "effective strategies for coordinating work, advancing technical skills and managing their reputation" [9]. Through this transparency, new types of leadership and influence may have emerged, but this has yet to be studied.

### 2.1. Social structure and leadership in open source software development

Studies of social structures and interaction in other working environments that offer transparency exist in the literature prior to GitHub's popularity. Here we highlight studies from the larger body of research on OSS development examining social interactions, through studies of online systems and how norms are developed among online teams (e.g. [10]).

Long and Siau [17] elaborate on "social structure" in SourceForge OSS projects by examining changes in communication network structures over time. They focus on interaction patterns in the bug tracking system in three SourceForge projects, showing that projects evolve from a single "hub" to a core/periphery structure. While these networks are centralized in the sense that a core group of individuals (or "key members") are at the center of communication, "it is decentralized in the sense that the decision or communication core is not concentrated on one or two members but a group of key members" [17].

Similarly, Crowston and Howison's [18] study of communications interactions in SourceForge's bug tracking system reveals the extent to which 120 project teams show uniformity and difference in their social structure. The communication structures surrounding bug fixes were neither consistently centralized nor decentralized. Instead, they display a range of centralization. One key pattern they uncover is that centralization is negatively correlated to both the number of developers and the number of active users contributing to bug reports. The authors offer their interpretation that as "projects grow, they have to become more modular, with different people responsible for different modules ... resulting in what might be described as a 'shallot-shaped' structure with layers around multiple [cores]" [18].

This multi-layered organizational structure was also reported in the study of Mockus, Fielding and Herbsleb [19] who found that, although development is quite centralized (with only 15 developers responsible for more than 80% of the code), centralization decreases dramatically in the context of new code contribution, bug fixes and bug reporting. Their findings lead them to this hypothesis: "In successful open source developments, a group larger by an order of magnitude than the core will repair defects, and a yet larger group (by another order of magnitude) will report problems" [19].

GitHub allows users to "follow" other individuals. The actions (commits, pull requests, comments, etc.) of followed users appear on a dashboard. This creates an additional layer to organizations of individuals who observe but do not participate. Due to the newness of GitHub's following feature, the social structure of followers has yet to be fully explored.

### 2.2. Following and followers on GitHub

Lee et al. studied a small set of GitHub users and examined how four users who are both the most popular and the most active (rockstars) influenced their followers [15]. They found that when rockstars are more active on the projects they own, they attract more of their followers to contribute to that project. They also saw that these rockstars guided their followers to new projects when they contributed on a new project. This study was limited to three months of activity of the four rockstars. Our study builds on these findings by studying all GitHub users with a large number of followers and considering all available contribution history. Further, our study considers the influence of all GitHub users with a large number of followers regardless of the amount they have contributed to projects on GitHub. This approach teases out the distinction between popularity and contribution. We also investigate who the most popular users are and the motivations behind following (or not following) others on GitHub.

*Following* is distinguished conceptually from contribution as a more passive act. On GitHub, Goggins and Petakovic [4] refer to following as affiliation, and pull requests and issue comments as participation. In their examination of influence across social technologies, they noted that different trajectories for influence exist among those who participate and those who affiliate with projects, groups or other forms of open online community. Using examples from across social technology platforms, Goggins and Petakovic illustrate how clear distinctions between acts of participation and acts of affiliation are important for understanding how influence occurs in social technologies.

A number of questions remain open. How does affiliation influence (follower influence) play out in a social coding environment like GitHub? How might follower influence be conceptualized as part of the previous participation-focused core/periphery model in OSS? The relationship between having a lot of followers, without consideration of activity level, and leading those followers in a particular direction is unexplored in prior studies of OSS in general, and GitHub in particular. In this paper we take a first step in this direction and tackle the following research questions:

*RQ1: Why do GitHub users follow others and who are the most followed users?*

*RQ2: Are GitHub users influenced by the users they follow?*

## 3. Method

To answer our research questions, we used a mixed-methods approach comprised of repository analysis as well as analysis of data from a survey of 800 GitHub users. The survey focused on the motivations behind following other GitHub users, while the repository analysis investigated the influence of such relationships. In this study, influence is defined as the ability of a user to guide their followers to star or contribute to new projects. In this section, we discuss our data sources, survey design, and process for analyzing survey responses. We focus our analysis on the actions of the *popular users* and their followers. We defined popular users as GitHub users who have many (500 or more) followers.

### 3.1. Description of survey

In order to understand why GitHub users utilize the ‘following’ feature, we conducted an online survey (see Appendix A) targeting three different user sets, (1) users who follow popular users, (2) users who follow other users but do not follow popular users and (3) users who do not follow any user on GitHub.

We emailed our survey to the 4000 most active GitHub users (measured by number of commits made in January to April 2014). We received 800 responses (20% response rate). All questions were optional (see Appendix A for number of responses per question). Survey questions were a mixture of multiple choice, open-ended and yes/no or agree/disagree. Firstly, we asked participants about their job, work-experience, their primary use of GitHub and the benefits of following users on GitHub. Then based on target user category, we asked them why they do or do not follow other users and how many users they follow. We also asked respondents if they follow specific user groups on GitHub, e.g. GitHub staff, creators of new library or framework, etc. and why they follow these specific user groups. We were also interested in knowing if respondents believe that the users they follow are experts.

**Table 1**
Occupation of survey respondents (in percentage).

| Occupation | Overall | Following popular users (1) | Following other users (2) | Not following (3) |
|---|---|---|---|---|
| Software developer | 87.6 | 92.8 | 86.2 | 83.7 |
| Manager | 14.6 | 16.4 | 12.4 | 15.3 |
| Student | 26.9 | 15.1 | 33.9 | 31.2 |
| Other | 8.4 | 6.2 | 8.1 | 12.1 |

**Table 2**
Survey respondents working on each type of GitHub project (in percentage).

| GitHub projects | Overall | Following popular users (1) | Following other users (2) | Not following (3) |
|---|---|---|---|---|
| OSS | 89.6 | 97.2 | 92.9 | 74.8 |
| Commercial (Proprietary) | 40 | 57.7 | 32.9 | 25.2 |
| Personal | 82.3 | 89.3 | 89.8 | 70.1 |
| Other | 7.1 | 4.8 | 7.4 | 9.8 |

**Table 3**
Inter-coder reliability. Krippendorff’s alpha scores.

| Question | Initial agreement | Final agreement | Alpha |
|---|---|---|---|
| What do you see as the benefits of following other people? | 0.87 | 1 | 0.94 |
| Why do you follow staff? | 0.74 | 1 | 1 |
| Why do you follow organizations? | 0.83 | 1 | 0.91 |
| Why do you follow contributors? | 0.81 | 1 | 0.93 |
| Why do you follow creators? | 0.82 | 1 | 0.92 |

### 3.2. Survey participants

We asked each participant to provide their occupation and the type of projects they work on within GitHub (multiple selections were allowed for each question). The majority of our survey respondents are software developers. There were also a significant number of managers and students. 35% of respondents selected more than one occupation. The most common mixed responses received were software developer and manager (14% of respondents) as well as software developer and student (19% of respondents). Table 1 shows the occupations of the respondents. The ’Other’ category responses included researchers, scientists, teachers, system administrators and designers.

Nearly 90% of our survey respondents use GitHub for OSS projects. Table 2 shows the type of project work our respondents use GitHub for. The ‘Other’ category includes respondents using GitHub for academic purposes (thesis, class-work, research, teaching, etc.), to store user configuration files, write books, for non-profit projects, presentations, transcribing and coding competitions.

### 3.3. Survey analysis

To analyze the responses of open-ended questions, we used methods from grounded theory [20]. We started with open-coding to define initial categories. We then performed axial coding to confirm that our open-codes represented the survey responses and to merge related categories. Two independent coders completed the coding. When each coder was satisfied with their codes, their code lists were combined to create a master code list. Each coder performed another iteration to apply a code from the master code list to each response. Next, the two coders discussed and reconciled any case where their codes did not align, resulting in 100% final agreement for all coded questions. We measured inter-coder reliability with Krippendorff’s alpha measure [21], results in Table 3. With the acceptance threshold

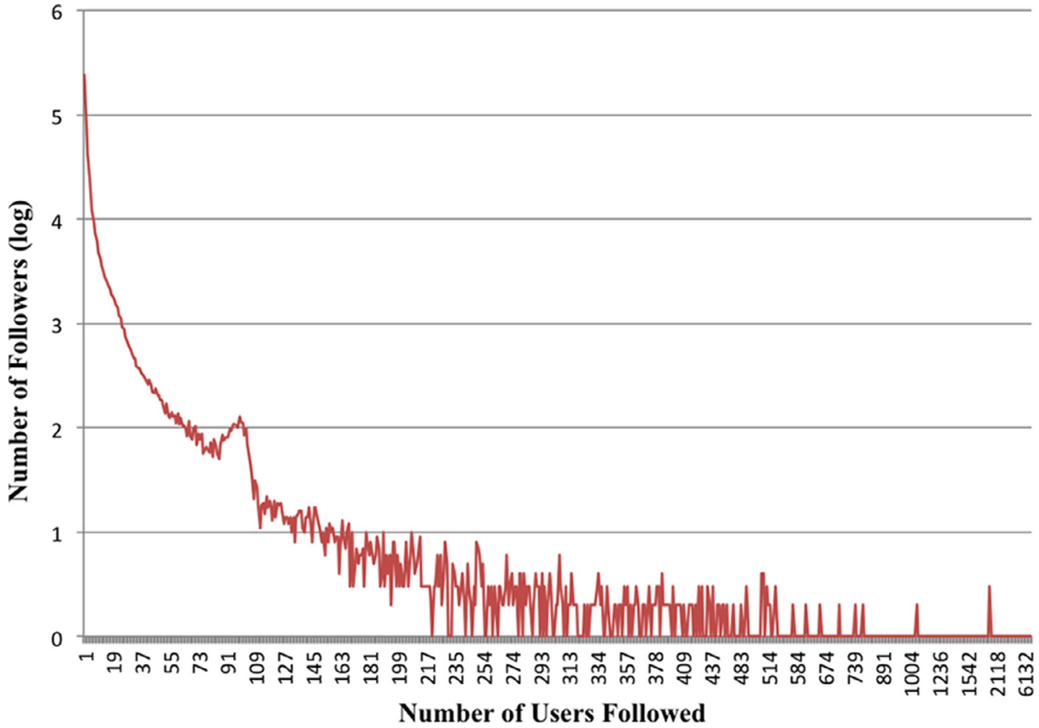


**Fig. 1.** Number of users the followers are following.

for Krippendorff's alpha at > 0.8, our scores indicate high inter-coder reliability.

### 3.4. Repository analysis

To investigate the activity of users who follow others and users who have many followers, we performed a repository analysis in GitHub. We obtained data from the GHTorrent project [22], which provides a mirror of the GitHub API data. The GHTorrent project fetches GitHub public data using the GitHub REST API and stores it in MongoDB and MySQL databases. We used the 2014-04-02 MySQL data dump, which included 8,510,504 projects and 3,426,046 users. The dataset contains 527,712 users who follow other users. For popular users, we used a threshold of 500 or more followers. There are 199 popular users, which is a manageable set of elite users to analyze. These 199 users have a combined set of 101,688 followers and have created, contributed to, forked or starred 49,927 projects.

## 4. Results

### 4.1. Motivation for following

*RQ1: Why do GitHub users follow others and who are the most followed users?* Following other users is popular on GitHub—more than half of a million GitHub users follow at least one other user. Fig. 1 shows the number of users being followed by GitHub users. Many users follow only a small number of other users. Of the users who do follow at least one other user, 45.5% follow only one user and 15% follow only two users. This suggests a core/periphery structure of popularity on GitHub with a small set of very popular users surrounded by a large set of their followers and other less popular users.

#### 4.1.1. Motivation behind following

In our survey, we asked all respondents to explain the benefits of following others on GitHub. Through axial coding, we categorized the responses into 8 categories:

- *Getting updates on activity.* 27.2% of respondents said the benefit of following others on GitHub is to receive updates on their activity. Even though GitHub also allows users to "watch" a project to receive notifications on project activity, some users stated that they would prefer to follow the main contributor or project owner to see updates on project activity. *"[Following others is] good for keeping track of how others are progressing with their own projects or how much they have contributed lately."*
- *Discovering new projects and trends.* 24.8% of the users surveyed follow other users with similar interests in order to identify new projects and stay up-to-date with the latest trends in OSS or a particular technology. *"If I find a skilled person that is active in the type of projects I'm interested in. By following that person I might stumble upon a new open source projects."*
- *No benefit.* 18.8% of total respondents (including those who do follow others) could not identify a benefit of following others. *"Not much value in following people. Far too noisy. Twitter is the only place I actively follow people and read anything."*
- *Learning.* 11.2% of respondents follow others to learn from them. *"It is a great opportunity to learn from their code and see how thing are done by the best in the business."*
- *Socializing.* 7.3% of respondents follow others for social reasons. They follow their friends and co-workers and/or influential developers to show respect or support. They also feel motivated by their followers. *"Showing a peer that you respect them via a follow. I don't use follows to keep tabs on what others are doing though I just do it for the socialization."*
- *Collaboration.* 5.8% of respondents follow others to identify opportunities for collaboration or to share code. *"If somebody is working on a similar problem to you, you can collaborate with them through pull requests to solve problems"* and *"Being able to grab commits from others repo."*
- *General interest/Miscellaneous.* 2.8% of respondents follow others out of general interest or for reasons that did not fit into any of our categories. *"see if something interesting created or changed."*
- *Gaining easy access to others.* 2.1% of respondents follow others to "bookmark" their profiles so they can easily access them later. *"It's*

*like bookmarking interesting people – I could potentially come back to their profiles"*

While these reasons may appear to mostly relate to the social network within GitHub, such social aspects can have an important impact on development activities [23]. For example, our results show that 24.8% of respondents indicate they follow others to discover new projects and trends. Thus, these social relationships guide potential new contributors to OSS projects who rely on these volunteers for their success. Furthermore, 11.2% indicate they follow others to learn; therefore, these relationships can develop the technical skills of emerging developers. For these reasons, the social network of following is impacting the working patterns within GitHub.

The stated benefits were consistent across all of the occupations of our survey respondents. While we saw some minor differences (for example, managers were more likely to state that following others provides no benefit with 21.2% of managers giving this response compared to only 13.8% of students and 18.9% of software developers), the differences were not statistically significant across the occupations for any of the categories.

### 4.1.2. Profiling popular users

There are 199 users who have 500 or more followers on GitHub. We examined each of these users' GitHub profile page and, if a link was available from their profile, their personal webpage. Through this analysis, we identified four categories of popular users: (1) GitHub staff, (2) organizations, (3) OSS developers, and (4) creators of library/frameworks.

It should be noted that before GitHub introduced their 'organizations' feature many companies and projects used a shared GitHub user account to represent their organization. Thus, organizations could be followed much like any other user. Once the GitHub organizations feature was introduced, many teams transferred their existing user account to an organization account. It is not possible to follow a GitHub organization account in the same way users are followed. Thus, organizations will no longer be a category of popular users once the GitHub organizations feature is fully embraced and all user accounts are transformed to utilize this feature.

Many, but not all, popular users are also very active. Table 4 shows the activity levels of the popular users who are not organizations. For many of the metrics, the median is much smaller than the mean showing that there is a wide range of activity levels among the popular users.

**Table 4**
Popular users' GitHub activity.

| | Mean | Median |
|---|---|---|
| Followers | 1348.6 | 891 |
| Projects starred | 91.7 | 38 |
| Forks | 28 | 15 |
| Pull requests | 68.7 | 21 |
| Commits | 3320.2 | 2203 |
| Comments | 1602.1 | 801 |
| Organizations (member of) | 2.7 | 2 |
| Projects (member of) | 20.6 | 14 |

**Table 5**
Survey responses to 'Do you consider the people you follow as experts?' (in percentage).

| Categories | Overall | Following popular users (1) | Following other users (2) |
|---|---|---|---|
| Yes | 39.54 | 44.03 | 33.9 |
| No | 8.47 | 4.11 | 12.44 |
| Maybe | 51.9 | 51.85 | 53.65 |

**Table 6**
Reasons for following popular user categories (in percentage).

| Categories | GitHub staff | Orgs | OSS devs | Creators |
|---|---|---|---|---|
| Obtain updates on activity | 31.6 | 57 | 32.3 | 45.3 |
| Find new projects/trends | 9.2 | 5.6 | 14.5 | 10.5 |
| No benefit | 1.3 | 0 | 1.6 | 1.1 |
| Learn | 10.5 | 5.6 | 9 | 7.5 |
| Socialize | 11.9 | 2.8 | 16 | 5.9 |
| Coordinate | 1.3 | 2.8 | 1.5 | 1.1 |
| General interest/Misc. | 9.2 | 2.8 | 9.3 | 6 |
| Easy access to others | 0 | 0.7 | 0.6 | 4.2 |

*Popular users are more likely to be viewed as experts.* We asked survey respondents if they believe that the users that they follow are experts. 39.54% of survey respondents responded positively that the users that they follow are experts, while only 8.47% believe that the users they follow are not experts. The remaining 51.9% respondents responded neutrally. Respondents following popular users were more likely to believe these users are experts ($x^2 = 5.1$, $p = 0.02$) and less likely to believe they are not ($x^2 = 10.98$, $p < 0.001$) than respondents who follow users who are not among the 199 most popular, as illustrated in Table 5.

### 4.1.3. Motivation for following popular users

We divided our survey respondents into three groups. Group 1 is composed of users who follow the 199 popular users who have 500 or more followers. Group 2 includes people who only follow users outside the 199 popular users, and Group 3 includes users who do not follow other users.

For survey respondents who did follow popular users, we asked whether they followed users in each of the popular user categories and why they followed those users. Table 6 shows reasons given by survey respondents for following popular users from these defined categories. We excluded any responses that did not provide a reason for following these users.

The most popular reason for following each of these user types is to obtain updates on activity. Finding new projects and trends was not cited as often for these users as it was for following in general.

### 4.1.4. Motivation behind not following

The responses from Group 3 respondents who do not follow any other users represent the perceived benefits of following others, while the responses from Groups 1 and 2 represent the actual benefits experienced by users who do follow others. Table 7 shows survey responses broken into the three groups. There are a number of differences in the responses across groups.

*Collaboration is more a perceived benefit than an actual benefit.* Interestingly, the users who do not follow any other users (Group 3) were more likely to cite collaboration as a benefit of GitHub's following feature than the users who did follow others (Groups 1 and 2). A chi-squared test of difference in proportion shows the difference is significant ($x^2 = 5.7$, $p = 0.017$). This could show that collaboration is only a perceived benefit of following, but the actual implementation of the feature falls short in this area. We used chi-squared tests in our analysis since our dataset was large and, thus, the expected value for each observational class was greater than 5 [24].

*Finding new projects/trends likely not a well-known benefit.* Respondents who do not follow any other users (Group 3), were much less likely to cite finding new projects/trends as a benefit of following others. A chi-squared test of difference in proportion shows the difference is significant ($x^2 = 55.7$, $p < 0.001$). This is not a well-advertised benefit of following, and it is likely that those who have never used

**Table 7**
Benefits of following users on GitHub. Percentage of responses from each targeted group (in percentage).

| Categories | Overall | Following popular users (1) | Following other users (2) | Not following (3) |
|---|---|---|---|---|
| Updates on activity | 27.2 | 23.2 | 30.5 | 29.4 |
| New projects/trends | 24.8 | 37.4 | 23.3 | 5.2 |
| No benefit | 18.8 | 14.7 | 11.5 | 37.3 |
| Learn | 11.2 | 10.0 | 13.1 | 10.5 |
| Socialize | 7.3 | 8.5 | 9.3 | 2.6 |
| Collaboration | 5.8 | 3.1 | 5.1 | 10.5 |
| General interest/Misc. | 2.7 | 1.2 | 5.1 | 3.3 |
| Easy access to others | 2.1 | 1.9 | 3.0 | 1.3 |

**Table 8**
Popular user influence. Followers who star, contribute to, or fork a project after a popular user stars, forks, contributes to, or creates a new project as their first activity on a project. The ratios in this table represent the average percent of followers for all popular users who performed each given action as their first action on a project over all projects. Popular users can take only a single type of action as their first action on a given project, and followers may follow more than one popular user. Therefore, the columns are not expected to add up to 100%.

| | Popular user first activity (in percentage) | | | |
|---|---|---|---|---|
| Follower first activity | Star | Fork | Contribute | Create new project |
| Star | 24.7 | 10.2 | 23.9 | 17 |
| Contribute | 12.5 | 6.7 | 13.7 | 3.3 |
| Fork | <0.1 | <0.1 | <0.1 | <0.1 |
| **Total** | **37.2** | **16.9** | **37.6** | **20.3** |

the follow feature simply have not thought about this as a possible benefit.

Not surprisingly, many respondents who did not follow any other GitHub users (Group 3) did not believe there was a benefit from following others (37.3%). We also asked respondents from this group why they did not follow others on GitHub. The reasons were varied:

- 39.5% found no benefit of following others on GitHub. *"I don't see the value in it."*
- 14.2% found watching projects more useful. *"I would rather follow the projects that they contribute to instead. I think following people is rather unnecessary a programmer works on specific projects, not with specific people."*
- 11.6% were not aware of the 'following' feature. *"Hadn't noticed the feature or found a need for it."*
- 7.9% use GitHub for personal projects. *"I just use GitHub for personal projects; I don't have an interest in the social features."*
- 6.8% are busy. *"Following someone else's code contributions sounds like a significant time investment to get value out of it. Right now, I don't have the time for that."*
- 6.8% were experiencing information overload. *"I already have too much information to follow. I wouldn't be able to keep up with receiving notifications about other GitHub users activities."*
- 3.2% respondents follow GitHub users on other social media or websites. *"I generally use twitter to follow the people/topics I am interested in."*
- Miscellaneous (10%).

Again, these reasons for not following others were consistent across each of the occupations of our survey respondents.

These findings have important HCI-related implications as social features continue to be added to software development environments. First, information overload is still an important problem and should be considered when implementing features that provide notifications of activity. Second, while the main purpose of following was to provide notifications of activity, many users are using these notifications to explore and find new projects within their areas of interest. Users who do not follow others do not seem to be aware of this benefit of the follow feature. Sites like GitHub could likely find ways to tailor notifications for this purpose and encourage this exploration.

### *4.2. Influence of following*

*RQ2: Are GitHub users influenced by the users they follow?*
To answer this research question, we examined user behavior. We performed a quantitative analysis of GitHub behavioral trace data to determine if the actions of popular users influenced their followers to star or contribute to new projects.

We analyzed the actions of the 173 popular users who are not organizations to identify evidence of influence in their followers. We examined when a popular user identifies a new project by creating, starring, forking or contributing to a project for the first time to see if their action attracted their followers to that project. We considered the following actions for project contributions: (1) making a commit; (2) submitting a pull-request; (3) creating an issue; or (4) commenting on an issue, commit, or pull-request. We analyzed the first activity of a popular user's followers who joined the same project after that popular user. We analyzed starring, contributing and forking for the followers' first activities.

*Popular user actions attract their followers to new projects.* We found that 43.5% of followers are attracted to new projects when a popular user whom they are following performs some activity on that project. Table 8 shows the first activity a popular user makes on a new project and the first activity of their followers who joined the same project after the popular user's activity on that project. Conversely, only 0.01% of a popular user's followers star or contribute to a new project before the popular user indicating that the popular user is influencing his followers and not just part of the herd. In fact, for users who are following the popular users, popular user activity on a project precedes the follower's activity for 90.5% of the new projects they star or contribute to.

*Followers are likely to star new projects after a popular user whom they are following does any activity on that project.* Table 8 illustrates that when popular users star, fork, contribute to or create a new project, a large percentage of their followers (24.7%, 10.2%, 23.9% and

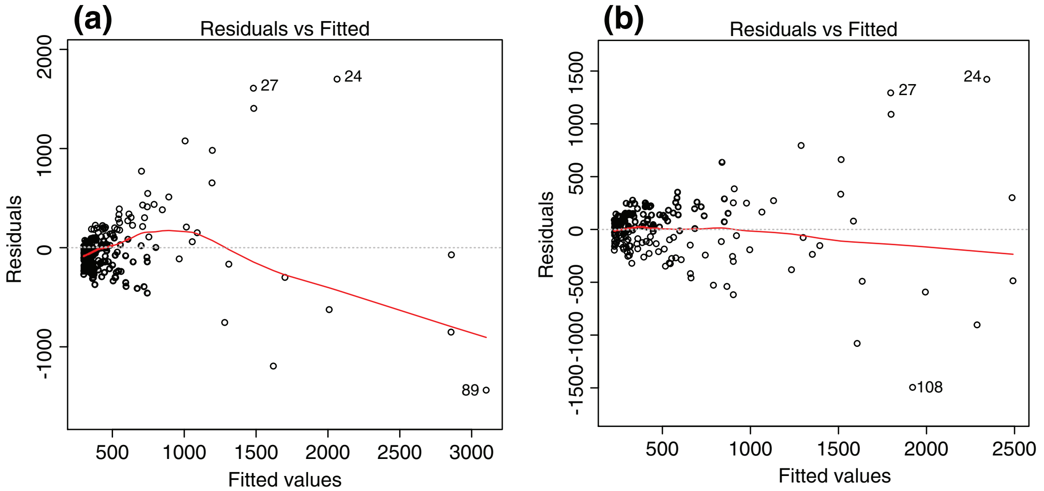


**Fig. 2.** Residuals vs. fits plots. (a) Linear model, (b) 5th degree polynomial model.

**Table 9**
Influence and popularity. Summary of the $R^2$ values using different degrees of polynomials.

| Degree | Predictive power ($R^2$) | Analysis of variance (compared to linear) |
|---|---|---|
| 1 | 0.607 | – |
| 2 | 0.683 | <0.001 |
| 3 | 0.688 | <0.001 |
| 5 | 0.694 | <0.001 |
| 10 | 0.754 | <0.001 |
| 15 | 0.789 | <0.001 |

17% respectively) will star that project. Considering that the most starred project on GitHub is starred by less than 1% of all users, these are in fact high percentages and suggest that the followers were influenced by the actions of the popular user.

Followers are also likely to contribute to new projects after a popular user whom they are following performs any activity on that project. As also shown in Table 8, the highest rate of contribution follows from a popular user contributing to or starring a new project with 13.7% and 12.5% of their followers contributing to that project.

*Starring a project is powerful.* We ran a Mann–Whitney test of distribution on the number of users influenced by a popular user when they star a project compared to when they contribute to a project. We expected that when a popular user contributed to a project they would influence more users than when they simply starred a project. However, a Mann–Whitney test shows no statistical significance. This demonstrates that contributions may not be any more powerful than stars in attracting users to a project. Mann–Whitney was selected to compare the average of the two groups since the data is not normally distributed.

*The more followers a popular user has, the more their followers are influenced by their actions.* There is a non-linear relationship between the number of followers a popular user has and the number of followers that they influence. We used curve fitting to identify the best fitting mathematical function to explain the relationship between influence (measured as the total number of a popular user's followers who star or contribute to a new project after the popular user performs some action on that new project over all projects) and the number of followers. A linear model ($Influence = NumFollowers$) was able to predict 60% of the data. However, as shown in Table 9, as the degree of the equation increases, so does the predictive power. We compared each polynomial equation to the linear equation with an Analysis of Variance and obtained significant *P*-values for each showing that polynomial functions are a better fit than the linear model. This shows that as a user's number of followers increases, their rate of influence increases.

Fig. 2 illustrates the residuals graphed against predicted values of *Influence* (fitted values) for both the linear and 5th degree polynomial statistical models. The residuals show the amount of variability in *Influenced* that is not explained by our model [25]. For the polynomial model (Fig. 2b), the shape of the line is flat except near the outliers, in contrast to the linear model (Fig. 2a), which is heavily distorted. This suggests a reasonable fit between the polynomial model and the data showing that the rate of influence increases as a person accumulates more followers.

The additional predictive power of the polynomial model follows our hypothesis that there is some point at which influence accrues faster than the number of followers alone. Our two statistical outliers, labeled as 24 and 27 in Fig. 2, further support this point. Those two popular users influenced significantly more actions than every other popular user, yet they do not have the most number of followers. These two users have very different profiles. User 27 is one of the most active popular users in terms of his number of comments on issues, pull requests and commits and his number of commits. He has created many popular OSS projects. While he has a large number of followers, he has less than half of the most followed popular user. On the other hand, user 24 is closer to average in regards to his activity within GitHub. However, he is one of the most followed users, and he is a member of a large number of organizations. He contributes to a wide range of projects.

*The more active a popular user is does not impact their influence.* There is not a clear relationship between the activity level of a popular user and the number of followers they influence. We measured the activity level of a user by considering various forms of activity, including number of commits, pull requests, forks, project membership, repositories owned and number of comments. Table 10 shows the results of a linear regression model that considers each of these measures of activity together with the number of followers a user has. The number of comments (made on issues, commits and pull requests) is the only measure of activity that has an impact on influence; this is interesting since this is the most social of our activity measures. However, even for this measure, the effect size is small as measured by Cohen's $f^2$. Conversely, the number of followers shows a large impact on influence. This indicates a social influence is occurring where followers are influenced by the popularity of a user more than the user's development activity.

**Table 10**
Impact of various measures of activity on influence compared with the influence of popularity (measured by number of followers).

| | *t*-value | Cohen's $f^2$ |
|---|---|---|
| Intercept | 0.124 | |
| Commits | −0.557 | 0.002 |
| Pull requests | −1.378 | 0.011 |
| Repositories owned | −1.187 | 0.008 |
| Forks created | 1.046 | 0.006 |
| Project membership | 1.684 | 0.016 |
| Comments | $4.99^{***}$ | 0.144 |
| Followers | $17.796^{***}$ | 1.831 |

($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$).

## 5. Discussion

Our research goal was to better understand the following relationship on GitHub. GitHub users can "follow" another user to receive notifications about that user's activity on GitHub. We investigated why users choose to follow other users and what types of actions are taken as a result of receiving these notifications. Our main goal was to identify if a user's actions are influenced by the notifications they receive through these following relationships.

*RQ1: Why do GitHub users follow others and who are the most followed users?* To answer our first research question, we surveyed 800 GitHub users and investigated GitHub's 199 most popular users (measured by number of followers). We identified four categories of popular users: (1) GitHub staff, (2) organizations, (3) OSS developers, and (4) creators of library/frameworks.

Getting updates on activity is the most commonly cited reason for following others showing that following others is mostly used an awareness mechanism. Another commonly cited reason for following others is to discover new projects and trends showing that users will investigate new projects that appear in their notifications. Other reasons given for following others were learning, socializing, collaboration and general interest. Users who do not follow others were more likely to cite collaboration as a benefit than users who do follow others, which suggests that collaboration, learning and socializing exist on a Continuum for GitHub users. Understanding their interconnection in future studies will likely help projects and companies who use GitHub to better understand how to build their project communities up by attending to a more sophisticated understanding of participant motivation.

Our findings show that following other users is being used for much more than project activity awareness. Yet, 14.2% of our survey respondents found watching projects more useful than following other users. When the goal of following is to maintain awareness around a particular project, watching the project of interest is intuitive since project activity is often much more than the actions of one contributor. This paper focused on how individuals are influenced by other individuals, and we, therefore, did not gather data or analyze project oriented following behavior. Future research could investigate influence of notifications from watching projects.

*RQ2: Are GitHub users influenced by the users they follow?* For our second research question, we analyzed the actions of popular GitHub users and their followers to look for evidence that popular users are influencing the actions of their followers. We found that when a popular user stars, forks, contributes to or creates a new project their followers are attracted to that project. We observed that a popular user's rate of influence increases as they accumulate more followers. However, their rate of contribution does not impact their rate of influence. This can indicate that popularity may be more influential than actual contribution. Thus, GitHub's following feature may be enabling a new type of leadership in GitHub-hosted OSS projects with popular users emerging as leaders. These findings can be validated by future studies. For example, a targeted survey could ask individual developers how they are influenced by particular popular users whom they follow. In Sections 5.1 and 5.2, we discuss our findings in relation to existing literature and their implications on OSS social structure and leadership, and we highlight open research questions our study identified.

### 5.1. New type of leadership emerging through following

Prior literature examined leadership in OSS projects as an explicit, participatory act of discussion or contribution evaluation [5–7,10]. In contrast to these participation-focused framings of OSS leadership, our findings show that a new type of leadership may be taking shape through follower relationships on GitHub-hosted projects. We found that popular users who are followed by many other users have influence over their followers, and thus are serving as leaders. The influence we observed was related to guiding users to new projects. Many OSS projects rely on volunteers to fix bugs and contribute new features, and having a large pool of potential contributors is important to the health of these projects. The fact that popular users are able to attract new contributors to projects is, therefore, important. If a project hosted on GitHub is able to attract a very popular user, it is likely that that user will bring a large pool of contributors with them leading to more contributions and greater project success.

The type of leadership we identified is, thus, at a higher-level than is typically studied as it over-arches all GitHub projects. Future research could focus on the impact of popularity within a project. *Does a popular user's opinion impact the success of a pull request? When a popular user comments on an issue or pull request does the sentiment of their comment influence the remainder of the discussion around that artifact?* It is also worth investigating the impact to a project when this type of leader leaves a project. *Do other popular leaders take their place? If not, does the project continue to thrive without a popular user leader?*

One type of popular user we identified were the creators of libraries/frameworks. For example, Linus Torvalds,[1] the creator of the Linux operating system, is one of the most popular users in our dataset. Yu et al. [26] found that project owners are followed by more users external to their project as their project grows in popularity. This indicates that very popular projects on GitHub will have at least one popular user. However, future research could investigate the existence of popular users and its relation to project success. Measures related to project success may be derived from the number of popular leaders engaged on the project or other metrics around popular leaders to build on existing measures of OSS project success [9,19,27–29].

Finally, the new type of leadership we have identified could have implications on OSS leadership theories for projects hosted on GitHub. For instance, we witnessed a very small subset of users who are very popular on GitHub indicating a new category of core/periphery structure in OSS. While Crowston and Howison [18] describe a shallot shaped structure focusing on contributions and work oriented leadership in OSS, our explication of popularity suggest a new type of leadership altogether. While popular users may also be contributors, they may not play the same central role in each particular project they contribute to. *Do popular users influence projects they participate in to a greater extent than the popularity they bring to a project from prior accomplishments in some cases? Do we need to re-conceptualize how we think about OSS project structure for projects hosted on GitHub?* These questions and others are important topics for future research. In OSS, as in team research in the physical world, we may find that task oriented and social influence are becoming two different ways of moving OSS projects forward.

[1] https://github.com/torvalds.

### 5.2. Popularity is altering OSS participation on GitHub

Further, we observed that a user's popularity impacts their level of influence. Prior literature on merit in OSS focused on contributions and technical skills [18,30]. People earn respect for getting work done. Our findings imply that popularity, now clearly visible in the GitHub interface, also influences how users perceive others. Marlow et al. [12] found that GitHub users use the information from user profiles to form impressions of each other and make judgments about potential contributors, which then influence whether or not their code contributions are accepted. *Are these impressions being made based on the users contributions and technical skills or on the user's popularity, which is prominently displayed on their profile? Are popular users' contributions more likely to be accepted on GitHub? Does a user's popularity reduce entry barriers on new projects?*

We found that as a user's popularity increases, so does their rate of influence. At some point, influence accrues faster than the number of followers alone. Investigating the most influential users did not reveal any particular characteristics of the users that would explain this phenomenon. Future research could continue this investigation. *Why are some users more influential than others? What are the characteristics of the most influential users?Are users more likely to follow already popular users?*

### 5.3. Threats to validity

One threat stems from our selection of the GHTorrent dataset, which may not be a full copy of all GitHub data [22]. Nevertheless, it is a best-effort approach that has been widely accepted in the research community as evidenced by its inclusion as the dataset for the MSR 2013 Mining Challenge [31] and the many recent papers that utilize its data in their analysis.

Another threat surrounds the dates in which a user began following another user, which may not always be accurate. This data is not available from the GitHub API [32], so this threat stems from a limitation of the GitHub API itself. GHTorrent attempts to obtain this data by monitoring GitHub events and recording the corresponding date. When GHTorrent was not able to record the corresponding event, it provides a best guess for this missing data by setting the following date to be the date of the creation of the latest created user of the follower or followee.

Our analysis is focused on all GitHub users with more than 500 followers. It is possible that our dataset is impacted by developers who have artificially increased the number of their followers through services such as the one available at http://githubfollowers.com. Such services typically result in an overnight spike in followers. We did not observe such spikes in our dataset, but the threat still exists that some users in our data are popular only because of this artificial measure. However, the existence of a small number of malicious users should not greatly impact our results.

Finally, our survey respondents were self-selected, and their opinions may not generalize to all GitHub users. However, we received a large number of survey responses (800) from a diverse set of users contributing to a wide-range of GitHub projects, and we reached saturation in our results.

## 6. Conclusion

GitHub's follow feature introduced a new social aspect to OSS projects where users can affiliate themselves with each other. We found that popular GitHub users attract their followers to new projects. As a user's popularity increases, so does their rate of influence. Our findings indicate that a new type of leadership may be emerging through follower relationships. Yet, activity level does not have the same impact on influence. A user's popularity is clearly visible to other users, thus users may be influenced by the popularity of a user showing a new social phenomenon emerging in OSS projects. Thus, the open collaboration environment of GitHub is shaping a new understanding of social structure on OSS projects shaped by popularity and influence.

This paper introduced several avenues for future research around popularity in software development projects. Future research should investigate the impact of popular users joining or leaving a project, the influence of a popular user within a discussion, and whether project success can be measured in some way based on popular users. Further, research can investigate popularity within transparent development environments to understand better if visible popularity itself elicits influence and attracts further popularity. Finally, research should continue to investigate how the social structure on OSS projects is changed by popularity.

## Acknowledgments

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada, as part of the NECSIS Automotive Partnership with General Motors, IBM Canada, and Malina Software Corp and the grant NSF and NSF VOSS OCI-1221254. The authors would like to thank the survey participants for the time they dedicated to this study.

## Appendix A. Survey questions and number of responses

1. Do you recognize yourself in any of the following categories?
   - Software developer (coder, tester, etc.)
   - Manager
   - Student
   - Other (please specify)
2. How much experience (in years) do you have in the occupation type that you selected in Question 1?
3. Do you use GitHub for any of the following work?
   - Open-source project
   - Commercial project
   - Personal project
   - Other (please specify)
4. What do you see as the benefits of following other people on GitHub?
5. Do you follow other people on GitHub? (Yes or No)

**Table A11**
Number of responses for each survey question.

| Question | Number of participants presented this question | Number of responses |
|---|---|---|
| 1 | 800 | 796 |
| 2 | 800 | 765 |
| 3 | 800 | 798 |
| 4 | 800 | 657 |
| 5 | 800 | 794 |
| 5a | 575 | 528 |
| 5b | 218 | 185 |
| 6 | 575 | 545 |
| 6a | 90 | 71 |
| 7 | 575 | 544 |
| 7a | 168 | 134 |
| 8 | 575 | 537 |
| 8a | 423 | 310 |
| 9 | 575 | 541 |
| 9a | 379 | 265 |
| 10 | 575 | 529 |
| 10a | 294 | 254 |
| 10b | 294 | 221 |
| 11 | 575 | 531 |
| 12 | 575 | 523 |
| 13 | 800 | 577 |
| 14 | 800 | 722 |

   (a) If yes, how many GitHub users do you follow?
   (b) If no, why don't you follow other people on GitHub? **(skip to Q13)**

6. Do you follow GitHub staff members on GitHub? (Yes or No)
   (a) If yes, why do you follow GitHub staff members on GitHub?
7. Do you follow organizations on GitHub? (Yes or No)
   (a) If yes, why do you follow organizations on GitHub?
8. Do you follow open-source contributors on GitHub? (Yes or No)
   (a) If yes, why do you follow open-source contributors on GitHub?
9. Do you follow creators of library, framework, technology, etc. on GitHub? (Yes or No)
   (a) If yes, why do you follow creators of library, framework, technology, etc. on GitHub?
10. Do you follow any other type of GitHub users? (Yes or No)
   (a) If yes, please specify what type of GitHub users?
   (b) If yes, why do you follow these other types of GitHub users?
11. Do you consider the people you follow as experts? (Yes, No or Maybe)
12. Do you follow GitHub users on other websites too? If yes, Please specify.
    - StackOverflow
    - Personal website/blog
    - Twitter
    - Facebook
    - Google Plus
    - Other (please specify)
13. What is your GitHub username? (optional)
14. Are you willing to participate in a follow-up survey or interview?

## References

[1] E. Bakshy, J.M. Hofman, W.A. Mason, D.J. Watts, Everyone's an influencer: quantifying influence on twitter, in: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, ACM, 2011, pp. 65–74.

[2] R.M. Bond, C.J. Fariss, J.J. Jones, A.D. Kramer, C. Marlow, J.E. Settle, J.H. Fowler, A 61-million-person experiment in social influence and political mobilization, Nature 489 (7415) (2012) 295–298.

[3] M. Cha, H. Haddadi, F. Benevenuto, P.K. Gummadi, Measuring user influence in twitter: the million follower fallacy, in: Proceedings of International Conference on Weblogs and Social Media, ICWSM, 10(10-17), 2010, p. 30.

[4] S. Goggins, E. Petakovic, Connecting theory to social technology platforms a framework for measuring influence in context, Am. Behav. Sci. 58 (10) (2014) 1376–1392.

[5] R.T. Fielding, Shared leadership in the apache project, Commun. ACM 42 (4) (1999) 42–43.

[6] A.J. Kim, Community Building on the Web: Secret Strategies for Successful Online Communities, Addison-Wesley/Longman Publishing Co., Inc., 2000.

[7] W. Scacchi, Free/open source software development practices in the computer game community, IEEE Softw. 21 (2004) 59–67.

[8] M. Michlmayr, Community management in open source projects, Eur. J. Inform. Prof. 10 (3) (2009) 22–26.

[9] L. Dabbish, C. Stuart, J. Tsay, 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, ACM, 2012, pp. 1277–1286.

[10] N. Ducheneaut, Socialization in an open source software community: a socio-technical analysis, Computer Support. Coop. Work 14 (4) (2005) 323–368.

[11] D. German, A. Mockus, Automating the measurement of open source projects, in: Proceedings of the 3rd Workshop on Open Source Software Engineering, Citeseer, 2003, pp. 63–67.

[12] J. Marlow, L. Dabbish, 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, ACM, 2013, pp. 117–128.

[13] J.Y. Moon, L. Sproull, Essence of Distributed Work: The Case of the Linux Kernel, in: P.J Hinds, S. Kiesler (Eds.), Distributed Work, MIT Press, 2002, pp. 381–404.

[14] W. Zhang, J. Storck, Peripheral members in online communities, in: Proceedings of Americas Conference on Information Systems, AMCIS, 2001, p. 117.

[15] M.J. Lee, B. Ferwerda, J. Choi, J. Hahn, J.Y. Moon, J. Kim, Github developers use rockstars to overcome overflow of news, in: Proceedings of Extended Abstracts on Human Factors in Computing Systems, CHI'13, ACM, 2013, pp. 133–138.

[16] J. Tsay, L. Dabbish, J. Herbsleb, Influence of social and technical factors for evaluating contribution in github, in: Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, pp. 356–366.

[17] Y. Long, K. Siau, Social network structures in open source software development teams, J. Database Manag. 18 (2) (2007) 25.

[18] K. Crowston, J. Howison, The social structure of free and open source software development, First Monday 10 (2) (2005).

[19] A. Mockus, R.T. Fielding, J.D. Herbsleb, Two case studies of open source software development: apache and mozilla, ACM Trans. Softw. Eng. Methodol. 11 (3) (2002) 309–346.

[20] J. Corbin, A. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Sage Publications, 2014.

[21] K. Krippendorff, Content Analysis: An Introduction to its Methodology, Sage, 2012.

[22] G. Gousios, D. Spinellis, Ghtorrent: Github's data from a firehose, in: Proceedings of the 9th IEEE Working Conference on Mining Software Repositories, IEEE, 2012, pp. 12–21.

[23] C. Amrit, M. Daneva, D. Damian, Human factors in software development: on its underlying theories and the value of learning from related disciplines; a guest editorial introduction to the special issue, Inf. Softw. Technol. 12 (56) (2014) 1537–1542.

[24] R.A. Fisher, F. Yates, Statistical tables for Biological, Agricultural and Medical Research, 3rd ed., Hafner Publishing Co., New York, NY, 1949.

[25] F.J. Anscombe, Graphs in statistical analysis, Am. Stat. 27 (1) (1973) 17–21.

[26] Y. Yu, G. Yin, H. Wang, T. Wang, Exploring the patterns of social behavior in github, in: Proceedings of the 1st International Workshop on Crowd-based Software Development Methods and Technologies, ACM, 2014, pp. 31–36.

[27] C. Fershtman, N. Gandal, The determinants of output per contributor in open source projects: an empirical examination, Available at SSRN 515282, 2004.

[28] P. Giuri, M. Ploner, F. Rullani, S. Torrisi, Skills and division of labor in an ecology of floss projects: implications for performance, in: Proceedings of the DRUID Tenth Anniversary Summer Conference, Copenhagen, Denmark, 2005, pp. 27–29.

[29] J.W. Kuan, Open source software as consumer integration into production, Available at SSRN 259648, 2001.

[30] W. Sack, F. Détienne, N. Ducheneaut, J.-M. Burkhardt, D. Mahendran, F. Barcellini, A methodological framework for socio-cognitive analyses of collaborative design of open source software, Comput. Support. Coop. Work 15 (2–3) (2006) 229–250.

[31] G. Gousios, The ghtorent dataset and tool suite, in: Proceedings of the 10th Working Conference on Mining Software Repositories, IEEE Press, 2013, pp. 233–236.

[32] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D.M. German, D. Damian, The promises and perils of mining github, in: Proceedings of the 11th Working Conference on Mining Software Repositories, ACM, 2014, pp. 92–101.