Measuring User Influence in Github: The Million Follower Fallacy

Ali Sajedi Badashian
University of Alberta
Edmonton, Canada
alisajedi@ualberta.ca

Eleni Stroulia
University of Alberta
Edmonton, Canada
stroulia@ualberta.ca

ABSTRACT

Influence in social networks has been extensively studied for collaborative-filtering recommendations and marketing purposes. We are interested in the notion of influence in Software Social Networks (SSNs); more specifically, we want to answer the following questions: 1) What does "influence" mean in SSNs? Given the variety of types of interactions supported in these networks and the abundance of centrality-type metrics, what is the nature of the influence captured by these matrics? 2) Are there silos of influence in these platforms or does influence span across thematic boundaries?

To investigate these two questions, we first conducted an in-depth comparison of three influence metrics, number of followers, number of forked projects, and number of project watchers in GitHub[1] (the largest code-sharing and version-control system). Next, we examined how the influence of the most influential software-engineering people in GitHub is spread over different programming languages.

Our results indicate (a) that the three influence metrics capture two major characteristics: popularity and content value (code reusability) and (b) that the influence of influentials is spread over more than one programming language, but there is no specific trend toward any two programming languages.

CCS Concepts

•**Software and its engineering** → **Software libraries and repositories;** General programming languages; •**Human-centered computing** → **Social network analysis;** Collaborative and social computing; •**Information systems** → Crowdsourcing;

Keywords

Mining software repositories, Software Social Networks, Programming languages, Social influence, Github

[1]http://www.github.com

CSI-SE'16, May 16 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2897659.2897663

1. INTRODUCTION

Recognizing the scope and magnitude of the influence of individuals in social platforms is important for value-added services on these platforms. For example, businesses may want to target their advertising to people with influence who may serve as "evangelists" for the advertised product or service to the community at large [16]. "Influencer marketing" as a means of advertising and marketing facilitates the transfer of information about products and services from an organization to its customers and partners [8]. It is even observed that voting patterns are influenced by postings in the Facebook feeds of the influential users [6].

This is why the notion of influence is now being widely studied in social networks [11, 9, 1, 26]. For the most part, this work is domain independent and does not attempt to recognize influentials in a specific discipline. We are interested in the study of influence in Software Social Networks (SSNs), and more specifically in GitHub, and how it is different/similar from regular social networks. We believe that focused studies of influence in different social platform are useful for domain-specific services. We observe that, currently, SSNs are expanding their service offerings. For example, Stack Overflow[2] launched Stack Overflow Careers in 2011, and Linkedin[3] provides job-search functionalities. Understanding the influence that an individual exerts in a specific software development community can help in utilizing those people for mobilizing information. There are many SSNs that propose professional recommendations and advertisements, based on the technical expertise of their users, their connections, and influence. For example, Stack Overflow recommends questions and also jobs for its users based on their achievements in different areas and their influence on the community (i.e., the up/down votes to their posts). This shows that the software community needs to pay more attention to the study of influence and influentials.

Since there are too many traces of work and communication in the SSNs, identifying the right practices, strategies and structures that are unique to a specific aspect of influence are challenging. As a result, researchers use a range of metrics to identify influencers [11]. Cha et al [9], in their highly cited "Million follower fallacy" paper, demonstrated that, in Twitter, users with many followers are not necessarily influential in terms of being retweeted or mentioned, but the most influential users can hold significant influence over a variety of topics. Dubois and Gaffney [11] studied influence in political parties and reported similar findings:

[2]http://www.stackoverflow.com
[3]http://www.linkedin.com

"number of followers" and "eigenvector centrality" measure traditional political status, while measures considering message quality (e.g., number of tweets containing a specific keyword) are indicators of content value.

In this paper, we report on a study of influence on GitHub, parallel to the studies of Cha et al. and Dubois and Gaffney, focusing on identification and comparison of influence metrics in GitHub. Furthermore, aiming to understand whether influence is thematic, we analyze the influence of GitHub members over different languages. Our work is driven by the following high-level research questions.

- What does influence mean in GitHub? Since this platform supports various types of interactions between the users, how might we interpret the various influence metrics derived from traditional centrality-type measures?

- How broadly does influence extend? Do the highly-reputed users exert influence over different communities? Or is one's influence focused within a theme, such as a specific programming language?

Our study revealed two interesting findings.

1. The numbers of "followers" of a GitHub member indicates his/her popularity and fame. On the other hand, the number of times the repositories of the developer have been "forked" is an indicator of the value of the content produced by the developer. The number of GitHub members "watching" the developer's GitHub activities captures a notion of influence between the above two indicators.

2. Influence is spread across different programming languages. In other words, influential GitHub members exert their influence in more than one programming languages. However, we found no specific trend over any two specific languages.

The rest of this paper is organized as follows. In the next Section, we review previous work in this area. Then, we introduce our data set in Section 3 and we analyze the indications of some influence measures in GitHub in Section 4. Finally, conclude the paper in Section 5.

2. LITERATURE REVIEW

2.1 Motivations Behind Studying Influence

Studying influentials is important in marketing [9]. Market researchers are interested in identifying the individual who is most able to spread an idea in a network [11]. This concept has an important role in dissemination of information like opinion propagation and viral marketing [13]. Influential people are valuable to businesses, with their power to convince their peers[2].

According to the "two-step flow hypothesis" (multi-step flow model), new ideas flow from the mass media to influentials, who, in turn, pass these ideas on to others that are more passive information seekers [14]. This, in practice, may reduce marketing costs for companies in spreading a product or technology. We specifically consider influence as the ability to cause desirable and measurable actions and outcomes in people [3, 16, 22, 15].

2.2 Influence in Social Networks

There have been a lot of studies targeting influence and influential people in social networks [26, 15, 19, 9, 11, 7, 29]. Sun and Vincent [26] define influential users based on the relationship between online posts, represented through a graphical model. Patil et al. [19] studied the trusted (friend of friend) and untrusted (enemy of friend) relationships in Epinions, the general customer review website. They found that the collective opinions of a user's friends regarding another user are strongly aligned with his own.

2.3 Implications of the Contributions of the Developers in SSNs

Focusing on software engineering, according to Storey et al. [24] a paradigm shift in social software engineering is emerging. This includes active engagement in online software development, which leads to high communication and fast diffusion of technologies in the communities. In fact, social media has changed the way developers collaborate [25]. Noting this, Storey et al. mentioned the roles of social media in collaborative activities at the team, project and community levels. They are not only forums, wikis and social networks that reflect these team-based activities, but also IDEs utilized mechanisms that integrate with online version control systems.

Dabbish et al. argue that a software project's success is influenced by the visibility of its developers' activities through motivating others [10]. In fact, developers choose the projects to which they contribute by considering the projects with frequent contributions or with contributions from developers with high status [27, 24].

According to Marlow et al. [18], in software social networks like GitHub, the transparency of the contributions enables users to form impressions of one's work. These impressions lead to future interactions and trust and utilization of someone's source code [24]. Pham et al. investigated this with GitHub users and concluded that this trust difference is evident in the ways that project managers treat the contributors of a developer they know and the developers they do not know. The later ones usually undergo more detailed examinations [20].

Marlow and Dabbish [17] interviewed a group of GitHub users (employers and job seekers) and discovered that employers obtain clues from developers' GitHub profiles (including activities, skills and interactions). The interesting fact was that these clues were viewed as more important than the developers' resumes. It is evident that, in open-source software engineering projects, the developers' participation in a project has sometimes a "value" higher than financial recompensation in commercial projects [17, 24]. The developers show their proficiency level in these contributions and spread their work record by the social clues available in GitHub. This is why Frèdèric Harper [12] –the senior technical evangelist at Mozilla[4]– mentions how to obtain influence and personal branding using online software social networks like GitHub and Stack Overflow. Growing the network and becoming more visible is possible through participation in these online repositories. Then, measures like reputation in Stack Overflow or number of followers in GitHub become indicators of an overall evaluation for these developers, which they manage and expand while performing their daily tasks.

[4]http://www.mozilla.org

2.4 Metrics of Influence

Blincoe, et al. [5] studied "following" as a metric of influence in GitHub and found that influentials guide their followers to new projects. They believe that, for the purpose of influencing others, popularity can be more important than contribution in projects. In another study, considering cross-references (links) between projects, Blincoe, et al. [4] identified software ecosystems. Considering these references as technical dependencies between GitHub projects, they argued that the users' social influence aligns with these technical dependencies within ecosystems. They found that popular ecosystems are mostly centered around tool support for software development. This is true for the "number of forked projects" popularity measure in our study.

Generally speaking, there are different measures and interpretations of influence in social networks. Cha et al. [9] considered three influence parameters; number of followers, mentions, and retweets of a user. They compared these metrics against each other and found that these metrics describe the influencer differently: being followed by numerous people indicates that one is a popular user (like public figures or news sources); numerous mentions in the tweets of others is an indicator of name value; finally, high levels of retweets of one's tweets indicates content value (e.g., content aggregation services or businessmen).

We are interested in studying influence in software social networks. In our previous work [23], we found that there is moderately high correlation (above 0.5) between developers' popularity, development and management characteristics in GitHub. This study aims for a deeper understanding toward popularity and influence metrics, to obtain a better indication for the available measures in software social networks. We attempt to study influence metrics in GitHub by replicating two important studies in this area [9, 11] regarding regular social networks and adding a variety of other analyses including influence analysis over different languages.

3. THE DATA SET

We used a MySQL database dump [21] (with a size of 25GB) containing information of 6,205,555 GitHub users and the meta-data regarding their social and development contributions. We consider "a GitHub account" equivalent to "a user". In fact, an account may be owned by an individual user, a company, or an open source library/platform. Since for some accounts, it is not exactly known if they are a company or a person, we prefer not to do personal judgments on filtering the accounts.

Then, for each user, u, we compute the cardinalities of three different sets: (a) the number of the user's followers, *u.follow*; (b) the number of distinct GitHub members who forked any one of the projects owned by u, *u.fork*; and (c) the number of GitHub members who watch any one of the projects owned by u, *u.watch*.

It is important to note here that we can compute all three measures only for the subset of the GitHub members who own at least one project; the *u.fork* and *u.watch* measures are not defined for users who do not have any projects associated with them. This subset includes about half of the GitHub membership, namely 3,524,603 users. A second interesting observation regarding this set is that, even though the measures are defined for all GitHub members, in a very large number of cases (one or more of) these metrics are 0, since many GitHub users have no followers and do not own projects "interesting" enough to watch or "useful" enough to fork.

We are motivated to consider these three metrics inspired by Cha et al.'s study [9]. Like many other networks, the *u.follow* measure is hypothesized to be an indicator of "popularity". *u.fork* is assumed to be related to usage and can be considered equivalent of the "retweets" in Cha et al.'s study. The *u.watch* metric can express interest and be considered equivalent to "mentions" in Twitter.

4. METHODOLOGY AND RESULTS

The methodology of our study is parallel to that of Cha et al. [9] and Dubois et al. [11] and it involves three steps.
Step1: We first analyzed the pairwise correlation of the three measures, using the Spearman correlation. The purpose of that step was to examine whether or not these three measures capture the same notion of "influence".
Step2: Next, we performed an in-depth qualitative analysis of the similarities and differences of the top 30 influential GitHub users according to each of the three measures. Our objective was to try to tease apart the differences in the profiles and behaviors of the three sets of influentials, corresponding to the three above measures.
Step3: Finally, considering the top 10 programming languages in GitHub, we calculated the *u.fork* and *u.watch* metrics over each language for all users. Next, we sorted the users based on their language-specific *u.fork* and *u.watch* measures, and we analyzed the pairwise correlation between the users' top two ranks (in the language-specific *u.fork* and *u.watch* influentials lists). This investigation enables us to understand whether a user's influence is focused on a single language or spread across multiple ones. Indeed, we high Spearman correlation indicates that the influential GitHub users tend to exert their influence across multiple (anywhere between two and eight) languages.

4.1 Correlations Bewteen Influence Measures

Table 1 reports the pairwise Spearman correlation between the three influence measures. The *u.follow* measure is only weakly correlated with the two other measures, while the *u.watch* measure is moderately correlated with *u.fork*.

Table 1: Spearman's rank correlation between influence metrics: P-value<0.01 (Significant at the 99% level)

Metrics	Correlation
u.follow vs *u.watch*	0.38**
u.follow vs *u.fork*	0.36**
u.watch vs *u.fork*	0.55**

In order to eliminate the possibility that these correlations are due to the large numbers of GitHub members with one or more "zero" values, we performed a more focused analysis, examining only the relatively influential users. We sorted the developers based on each of the three measures and we identified three different sets of top users based on each measure: the top 10% and top 1% GitHub users as well as the top 100 users. We then calculated the pairwise Spearman correlations between the three measures, which are reported in Table 2. The correlations between *u.follow* and the two others are mostly weak or non-significant, which

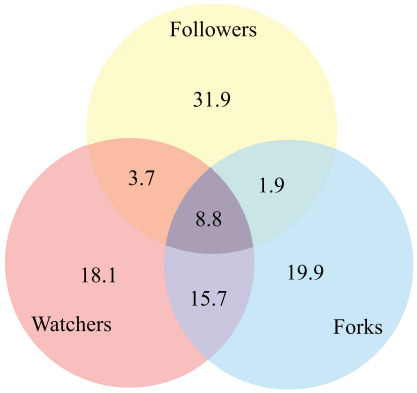

Figure 1: The top-100 influentials across measures, shown in a venn diagram: the values are normalized so that the total is 100%

implies that *u.follow* is independent from the two others and captures different notions of "influence" than the two others. *u.watch* and *u.fork*, however, are in most cases strongly correlated.

The three top 100 lists include 216 distinct accounts (that can be either individuals, companies or libraries) as shown in Figure 1. Note that the values depicted in the Figure are percentages over these 216 people. Interestingly, there is only a marginal overlap between *u.follow* and two other top lists (around 11% and 12%). The *u.watch* and *u.fork* lists however are sharing some more people (still just around 24%). This finding enforces the indication of the first step that the *u.follow* measure is different from the two others, but that the *u.watch* and *u.fork* are moderately correlated.

4.2 Looking into The Profiles of Influentials

In order to gain an insight on the nature of each metric, we took a deeper look on the top users' profiles in each category. We started investigating 30 top accounts in the three lists by checking their GitHub profiles. We examined their roles and appearance on the web (i.e., their personal page, company contributions, Wikipedia and social networks like Twitter). These roles are the ones that are directly mentioned in at least one of the resources in the web. We classified these roles in seven categories as it is shown in Table 3.

All the accounts in the top 30 list based on the *u.follow* influence metric are owned by individuals. Many of them are public figures, such as Linus Torvalds, Chris Wanstrath, and PJ Hyett (Co-founders of GitHub). There are also other enterprise roles (e.g., David Heinemeier Hansson; Founder and CTO at BaseCamp) or mixed roles (e.g., John Resig; lead developer of jQuery JavaScript library; Entrepreneur and Dean of Computer Science at Khan Academy). There is no "ORG" account in this category. We found that the focus is on the public figure in this category and conclude that this metric measures popularity the most. These are famous and popular people in software engineering (not companies or organizations), but due to the nature of GitHub, only the roles of active developers are highly being followed and more managerial roles are not seen here.

Considering the top 30 accounts based on *u.forks*, however, completely different findings were obtained. In this set, there are more "ORG" than "USR" accounts. More specifically, 25 accounts in the top 30 based on *u.fork* are organizations, usually big software companies (i.e., Facebook, Twitter, GitHub, Mozilla, Heroku, Apache, Google, Udacity, etc.), and 13 of these 25 are specifically libraries and frameworks (e.g., jQuery, Bootstrap and Angular-ui). On the other hand, only 3 of the top 30 in this category are developers or engineers. We argue that this phenomenon is not a byproduct of personal preferences: forking implies' code usage and content propagation and indicates "content value". In fact, they are influentials in that they produce reusable code which is adopted by communities of developers.

Similarly, less than half of the top 30 accounts based on *u.watch* measure are tagged with "ORG" which are big companies (e.g., Google, Facebook, GitHub, Adobe, Yahoo, etc.) or libraries and frameworks (e.g., jQuery and elasticsearch). However, in this category, there are also developers (e.g., Kenneth Reitz who is Python Overlord and Evangelist at Heroku or Sindre Sorhus who is a GitHub developer) and some entrepreneurs (e.g., James Halliday; founder of Substack, or, Victor Felder, Co-Founder of Kwak.io). Comparing with *u.fork*, there are fewer organizations and libraries in this group, but more engineers and developers that shows a semi-personal process. We interpret this group as a combination of the two others so that it indicates both popularity and code reuse. We can relate this group generally to "interest" of the users toward a project.

These results are similar to Cha, et al.'s study: "The most connected users are not necessarily the most influentials" [9]. As a result, we found similarities and differences with that study as are indicated in Table 4. In Twitter, the three metrics indicate three different measures (that respectively account for public figures, celebrities and content value). However, in GitHub, due to the technical aspects of the network, the measures are limited to two: "number of followers" accounts for popularity and fame, and, "number of forked projects" indicates code usability; "number of watched projects" is somewhere in between the two, but more connected to "number of forked projects" and can be interpreted as "interest".

Like Twitter, having many followers doesn't necessarily mean being actively influencer in action, although it can be in some cases, due to the technical nature of GitHub. There are better measures for real influence and impact which are number of forked projects and number of watched projects influence metrics. Between these two, however, the number of forked projects is more practical and genuine since shows actual code usage.

4.3 Influence over Different Languages

In our study of the 10 different programming languages, we focused on the top 1% followed users (37,022 users) for whom all measures were non-zero. We want to see if users are influential in the context of a single language or they are influential in several languages. First, we obtained the top 10 programming languages (see Table 5). The projects associated with these languages are around 85% of the total projects in GitHub with a specified, dominant language. Then, since number of followers is not project dependent, we ignored this metric and simply obtained the users' number of watched and forked projects over each of then 10 considered programming languages.

Figure 2 gives an idea of the "number of forked projects"

Table 2: Spearman's rank correlation between influence metrics for top users:
** P-value<0.01; significant at the 99% level
* P-value<0.05; significant at the 95% level
- P-value>0.05; non-significant results

	Top 10%			Top 1%			Top 100 users		
Sorted based on: →	*u.follow*	*u.watch*	*u.fork*	*u.follow*	*u.watch*	*u.fork*	*u.follow*	*u.watch*	*u.fork*
u.follow vs *u.watch*	0.49 **	0.40 **	0.49 **	0.52 **	0.23 **	0.33 **	0.23 *	0.17 -	0.39 **
u.follow vs *u.fork*	0.46 **	0.39 **	0.40 **	0.52 **	0.26 **	0.21 **	0.27 **	0.26 **	0.13 -
u.watch vs *u.fork*	0.75 **	0.70 **	0.67 **	0.91 **	0.67 **	0.64 **	0.86 **	0.63 **	0.30 **

Table 3: Different roles the top users (based on each measure) have

Role / Based on measure	Engineer, developer or designer	CEO, founder, co-founder, inventor or entrepreneur	Speaker	Author or writer	Education; professor	Open source library, framework, platform, etc.	Organization
u.follow	**25**	**12**	**4**	**7**	2	0	0
u.watch	**11**	**5**	0	0	1	**8**	**18**
u.fork	3	2	0	1	2	**13**	**25**

and "number of watched projects" measures over different languages. It shows how many users own projects in zero to ten programming languages that are being watched or forked by at least one other user. Most users are influential over fewer than four different languages, but there are some users who are influential over 8, 9 or even all 10 considered languages.

We found that, on average, each user has almost one more watched project than forked project (by another developer in the community). On average, each user has "forked projects" over 1.7 different programming languages (median=2) and "watched projects" over 2.5 different languages (median=3). This is not surprising because users tend to start by "watching" projects, and then proceed to "fork" them; developers choose which projects to fork based on the notifications they get from the projects they are watching.

There is a strong correlation (between 0.64 and 0.84) between number of forked projects and number of watched projects within each language. This alignment is similar to the results of Table 2.

An important outcome of the language study is to examine the breadth of influence over different programming languages as software communities or themes.

So we examined the correlation between the number of forked (or watched) projects over different languages and observed that there is no correlation between these measures in any pair of languages (all pairwise correlations are less than 0.3). This means that there is no pattern of influence across any two specific languages. But this does not reject the hypothesis that the users' influence spans across different languages. In order to investigate the breadth of influence of the users over different languages (without limiting it to a specific language for all the users), we obtained the rank of each individual in all the 10 languages (regarding number of watched/forked projects) separately. Then, disregarding the language, we prepared a list of all the 10 ranks of each user (if the user has an influence over that language). We then obtained the correlation between the top two ranks each user obtained over the 10 languages. We found that there is a moderate correlation between the top two languages of the users (0.51 regarding number of forked projects and 0.42 regarding number of watched projects measure). There is also a weak correlation between the top and the subsequent languages of each user (until the eighth language).

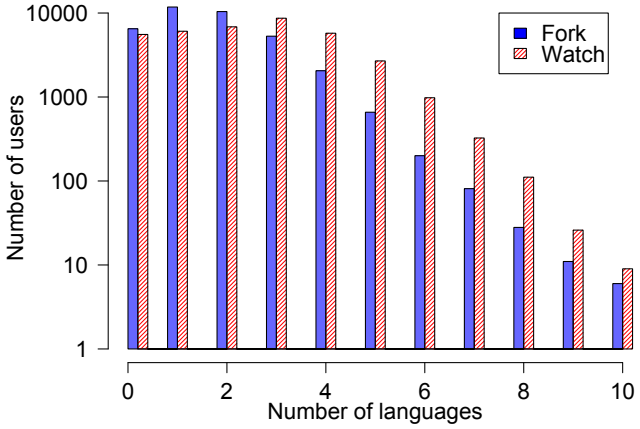

Figure 2: breadth of influence over different programming languages

These results indicate that the users' influence is spread over at least two different languages; there is no trend between any two specific languages, each user exerts influence of his/her own particular pair of languages. In other words, as a characteristic of the user, his/her influence is spread over different programming languages.

We also notice that "number of forked projects" influence is spread over more languages than the "number of watched projects" measure (although people have higher "number of watched projects" than forked ones). This indicates that the "number of forked projects" is not language-dependent but, rather, developer-dependent.

5. CONCLUSIONS AND FUTURE WORK

In this paper we examined three measures of a developer's influence in GitHub; the number of his/her followers, the number of watchers of the projects he/she owns, and the number of forks of the projects he/she owns.

- Users with many followers are not necessarily influ-

Table 4: Comparing influence metrics in Twitter and GitHub

Twitter[9]			GitHub		
Metric	Top users in the list	Indication	Metric	Top users in the list	Indication
follow	public figures and news sources	popularity; audiences are directly related to these influentials	*u.follow*	public but professional figures	popularity and fame in software engineering
mention	celebrities	name value	*u.watch*	public professional people, big companies and owners of libraries/frameworks	interest; both popularity and code usability
retweet	content aggregators, news sources and businessmen	content value	*u.fork*	big companies, owners of libraries/frameworks	code usability (content value)

Table 5: Top 10 programming languages considered

Language	# of projects
JavaScript	2,681,228
Java	1,150,381
Ruby	1,279,018
Python	1,112,088
PHP	987,731
CSS	588,552
C	587,579
C++	523,479
Objective-C	471,241
C#	339,779

ential in terms of how frequently their projects are watched or forked. The number of one's followers is a sign of popularity and fame in software engineering. The number of forks of one's projects is a sign of code value and reuse. Compared to Twitter, the number of followers and the number of one's projects forks are similar, in nature, to the follower and retweets counts. There is no Twittermeasure similar to the number of one's projects watchers; this measure indicates both popularity and content value, with a focus on the later; more generally, it can be viewed as a measure of "interest" as well.

- As a characteristic of the developer, influence is spread over different languages. We found that the users' influence is spread over two or more different languages, but there is no specific trend for alignment between any two programming languages.

These results are similar to the results of Cha et al. [9]. In Twitter, "having a million followers does not always mean much" [9]. Instead, having an active audience who retweet or mention the user is more influential. In GitHub, we found many successful software-engineering people with high number of followers. Here, due to the nature of GitHub –which is technical and development based– being followed means something more than Twitter; the highly followed people are not politicians and athletes, but professional software-engineering people who are more popular and esteemed by others (we validated this by both investigating the top 30 accounts and considering the weak correlation between *u.follow* and *u.fork*). So there is a subtle difference between the interpretation of "follow" based influence in GitHub and Twitter. Having said this, as the main conclusion, we found that like Twitter, the number of followers is not the best way to identify the actively engaging influencers. There are other measures that measure real influence in terms of impact and influence (e.g., , number of forked projects and number of watched projects in GitHub).

Also Cha et al. mentioned that in Twitter, the influence of most influentials is achieved by continuous work over different topics. Interestingly, we observed that in GitHub, users are influential over two or more programming languages, even though being active in several programming languages (in GitHub) is much more difficult than being an active twitter in various topics (in Twitter).

In the future, we plan to examine more potential indicators of influence like starred projects, social influence (e.g., commenting) and bug report relations (e.g., bug reporter and fixer) can help gain a better categorization of influence and popularity metrics. Cross-referencing with the study of pull requests and their acceptance [28] is another interesting study. We believe that recognizing the meaning of these metrics may help developers build their reputation while coding and transfer it to sustainable "digital-career" platforms for their future employment.

In general, the meaning and indications of influence in SSNs –as it was asked in our first research question– can be further investigated. The wider notion of influence can be explored using more in-depth behavioral analysis of the users (including developers). In this paper, we just addressed influence in a specific domain, which is directly related to **programming** and **pure source code development** in open-source communities as part of **software engineering** community. Other aspects, including software design, project management and software research, are critical in studying influence in software engineering community. Finally, in the long term, we plan to conduct inter-network analysis to identify the similarities and differences between general social networks and the SSNs.

6. REFERENCES

[1] I. Anger and C. Kittl. Measuring influence on twitter. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, pages 31:1–31:4, New York, NY, USA, 2011. ACM.

[2] M. Aziz and M. Rafi. Identifying influential bloggers using blogs semantics. In *Proceedings of the 8th International Conference on Frontiers of Information Technology*, page 7. ACM, 2010.

[3] F. M. Bass. A new product growth for model consumer durables. *Manage. Sci.*, 50(12 Supplement):1825–1832, Dec. 2004.

[4] K. Blincoe, F. Harrison, and D. Damian. Ecosystems in github and a method for ecosyste m identification using reference coupling. In *Proceedings of the 12th Working Conference on Mining Software Repositories*, MSR '15, 2015.

[5] K. Blincoe, J. Sheoran, S. Goggins, E. Petakovic, and D. Damian. Understanding the popular users: Following, affiliation influence and leadership on github. *Information and Software Technology*, 70:30–39, 2016.

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

[7] P. Bouros, D. Sacharidis, and N. Bikakis. Regionally influential users in location-aware social networks. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 501–504. ACM, 2014.

[8] D. Brown and N. Hayes. *Influencer Marketing: Who really influences your customers?* Routledge, 2008.

[9] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring user influence in twitter: The million follower fallacy. *ICWSM*, 10(10-17):30, 2010.

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

[11] E. Dubois and D. Gaffney. The multiple facets of influence identifying political influentials and opinion leaders on twitter. *American Behavioral Scientist*, 58(10):1260–1277, 2014.

[12] F. Harper. *Success in Programming: How to Gain Recognition, Power, and Influence Through Personal Branding*. Apress, 2015.

[13] G. Katsimpras, D. Vogiatzis, and G. Paliouras. Determining influential users with supervised random walks. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 787–792, Republic and Canton of Geneva, Switzerland, 2015. ACM.

[14] E. Katz. The two-step flow of communication: An up-to-date report on an hypothesis. *Public opinion quarterly*, 21(1):61–78, 1957.

[15] E. Katz and P. F. Lazarsfeld. *Personal Influence, The part played by people in the flow of mass communications*. Transaction Publishers, 1955.

[16] E. Keller and J. Berry. *The influentials: One American in ten tells the other nine how to vote, where to eat, and what to buy*. Simon and Schuster, 2003.

[17] J. Marlow and L. Dabbish. Activity traces and signals in software developer recruitment and hiring. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 145–156. ACM.

[18] J. Marlow, L. Dabbish, and J. Herbsleb. Impression formation in online peer production: activity traces and personal profiles in github. In *2013 conference on Computer supported cooperative work, 117–128*. ACM.

[19] A. Patil, G. Ghasemiesfeh, R. Ebrahimi, and J. Gao. Quantifying social influence in epinions. In *Social Computing (SocialCom), 2013 International Conference on*, pages 87–92. IEEE, 2013.

[20] R. Pham, L. Singer, O. Liskin, F. Figueira Filho, and K. Schneider. Creating a shared understanding of testing culture on a social coding site. In *Software Engineering (ICSE), 2013 35th International Conference on*, pages 112–121. IEEE, 2013.

[21] The GHTorrent Project. Mysql database dumps, data dump of 2015-04-01. "http://ghtorrent.org/downloads.html", Visited on 2014/08/20.

[22] E. M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[23] A. Sajedi Badashian, A. Esteki, A. Gholipour, A. Hindle, and E. Stroulia. Involvement, contribution and influence in github and stack overflow. In *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, CASCON '14, pages 19–33, Riverton, NJ, USA, 2014. IBM Corp.

[24] M.-A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky. The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*, pages 100–116. ACM, 2014.

[25] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 359–364. ACM, 2010.

[26] B. Sun and V. T. Ng. Identifying influential users by their postings in social networks. In *Proceedings of the 3rd International Workshop on Modeling Social Media*, MSM '12, pages 1–8, New York, NY, USA, 2012. ACM.

[27] C. Treude and M.-A. Storey. Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. In *Software Engineering, 2010 ACM/IEEE 32nd International Conference on*, volume 1, pages 365–374. IEEE, 2010.

[28] J. Tsay, L. Dabbish, and J. Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *36th international conference on Software engineering*, pages 356–366. ACM, 2014.

[29] C. Zhou, P. Zhang, J. Guo, and L. Guo. An upper bound based greedy algorithm for mining top-k influential nodes in social networks. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 421–422. ACM, 2014.