Birds of a Feather Flock Together? A Study of Developers' Flocking and Migration Behavior in GitHub and Stack Overflow

Sandeep Kaur Kuttal [#1], Michael Mu Sun [#2], Akash Ghosh [#3], Rajesh Sharma [*4]

[#] *Tandy School of Computer Science, University of Tulsa*
USA
[1] sandeep-kuttal@utulsa.edu
[2] sun@utulsa.edu
[3] akashghosh@utulsa.edu

[*] *University of Tartu*
Estonia
[4] rajesh.sharma@ut.ee

***Abstract*—Context: Interactions between individuals and their participation in community activities are governed by how individuals identify themselves with their peers. Although software developers collaborate using online peer production sites, this phenomenon has not been studied across online peer production sites in software engineering. Knowledge of this may help tool builders and researchers gain better insights about developers' expectations for online peer production sites.**

Objective: We want to investigate such behavior for developers while they are learning and contributing on socially collaborative environments, specifically code hosting sites and question/answer sites. In this study, we investigate the following questions about advocates, developers who can be identified as active learners and well-rounded community contributors. Do advocates flock together in a community? How do flocks of advocates migrate within a community? Do these flocks of advocates migrate beyond a single community?

Method: To understand such behavior, we identified 12,578 common advocates across a code hosting site - GitHub and a question/answering site - Stack Overflow. These advocates were involved in 1,549 projects on GitHub and were actively asking 114,569 questions and responding with 408,858 answers and 1,001,125 comments on Stack Overflow. We performed an in-depth empirical analysis using social networks to find the flocks of advocates and their migratory pattern on GitHub, Stack Overflow, and across both communities.

Results: We found that 7.5% of the advocates create flocks on GitHub and 8.7% on Stack Overflow. Further, these flocks of advocates migrate on an average of 5 times on GitHub and 2 times on Stack Overflow. In particular, advocates in flocks of size two migrate more frequently than larger flocks. However, this migration behavior was only common within a single community.

Conclusions: Our findings indicate that advocates' flocking and migration behavior differs substantially from the ones found in other social environments. This suggests a need to investigate the factors that demotivate the flocking and migration behavior of advocates and ways to enhance and integrate support for such behavior in collaborative software tools.

GitHub; Stack Overflow; Social Network Analysis; Flocking; Migration; Developers; Advocates.

I. Introduction

"Birds of a feather flock together." William Turner introduced this phrase, which highlights that individuals with similar identities create flocks (communities) and tend to migrate (participate) as a group between community activities [52], [31], [30]. Researchers from different backgrounds have found compelling evidence of this phenomenon in the domain of politics [35], scientific research [54], medical tests [24], friendship among adolescents [57], and email spammers [22]. Similarly, in software engineering, this phenomenon has been observed as software developers collaborate (flock) using online peer production sites. However, the occurrence this phenomenon across online peer production sites has not been studied. The knowledge of this may help tool builders and researchers gain better insights about developers' expectations for online peer production sites.

Developers with technical skills are competent in programming and writing quality code, while those with social skills are intrinsically passionate/motivated coders, good collaborators, and project managers. A developer exhibiting both sets of skills is known to possess "socio-technical skills"[50]. Socio-technical skills are profoundly appraised by developers to evaluate new team members [13], [12] and by managers to make hiring decisions [50], [38], [42], [44]. Aware of this fact, software developers demonstrate their code contributions through code hosting sites and their learning through technical Questions and Answers (Q&A) sites [23], [60]. The most popular site for code hosting is GitHub [11], [55] and for technical Q&A is Stack Overflow [41].

GitHub is a collaborative site for large scale software development that facilitates code sharing and version control. Some activities on GitHub include forking projects (creating a copy), contributing code, and collaborating globally [55]. Developers' behavior on GitHub is influenced by the fact that they are being observed by their peers [26] and hiring managers [50]. GitHub profiles and projects can provide enriched logs regarding the socio-technical skills of a developer. For example, logs may include what projects a developer owned and contributed to, the quantity of code they produced, the diversity of the languages they used, and the time they took to finish the project [43], [50], [42], [48], [59].

Stack Overflow is a community-based website for asking and answering questions related to programming languages, software engineering, and software tools [10]. It facilitates code reuse by providing better solutions [45] in less time [41]. Hence, developers active on Stack Overflow hone their socio-technical skills by seeking advice and help from their peers [45] and by sharing their knowledge and expertise to educate others. Developers are motivated to contribute more [28], as potential recruiters use Stack Overflow profiles to determine their expertise [21]. Stack Overflow logs provide histories about socio-technical skills such as quality of answers, reputation in the community with scores, badges, and votes [48].

GitHub and Stack Overflow create an ecosystem for developers to share their knowledge. Code hosting sites like GitHub display developers' passion for developing software while technical Q&A sites like Stack Overflow reflect their altruistic nature to help the community to learn. The developers who are active in both communities can be identified as considerate, hobbyist, and protean coders. Vasilescu et al. [58] observed that developers who were more active on GitHub (higher number of commits) acted as "teachers" (answered more questions) to the individuals on Stack Overflow. Furthermore, Lee et al. [36] found that developers share common interests when they are involved in the same activities on GitHub and Stack Overflow. Hence, developers sharing common interests and that are active in both GitHub and Stack Overflow sites can be considered as a flock. We refer to these individuals as **"advocates"** throughout our paper.

An advocate has to be exceptional in technical as well as social skills. The activity traces of an advocates' socio-technical skills are distributed across multiple projects on GitHub and Q&As on Stack Overflow. These traces can help us understand how advocates contribute across projects and production sites.

To understand the flocking and migration of advocates active on both peer production sites, we formulated the following research questions:

- RQ1: Do advocates flock on a peer production site?
  - How do the advocates tend to flock?
  - What characteristics motivated advocates to flock?
- RQ2: Do the flocks of advocates migrate within a peer production site?
  - How do the advocates migrate within the sites?
  - What characteristics motivate a flock of advocates to migrate?
- RQ3: Do the flocks of advocates migrate beyond a single peer production site?
  - What characteristics motivate a flock to migrate across sites?

To answer the above research questions, we collected 12,578 common advocates across a code hosting site - GitHub - and a question/answering site - Stack Overflow. These advocates were involved in 1,549 projects on GitHub and were actively asking 114,569 questions and responding with 408,858 answers and 1,001,125 comments on Stack Overflow. By conducting social network analysis, we found that 7.5% of the advocates created flocks on GitHub and 8.7% on Stack Overflow. Further, these flocks of advocates migrate on an average of 5 times on GitHub and 2 times on Stack Overflow. In particular, advocates in flocks of two migrate more frequently than larger flocks. However, this migration behavior was only common within a single community of GitHub or Stack Overflow.

II. Social Network Terminologies

The field of social networks is based on graph theory, which treats individuals as nodes and the connections among these individuals as edges SNA. Let $G(N,E)$ be an undirected graph, where N represents all the users of an online peer production site under investigation, and E as the set of interactions among the users. Two users n_i and n_j in a Graph G, are connected if they have interacted on the online peer production sites. The following are the social network terminologies used in this paper:

- **Degree:** Let $\aleph_i$ represent a set of nodes connected to a node $n_i \in N$. So, $|\aleph_i|$ represents the degree of the the node n_i. Each connected node in the set $\aleph_i$ is also referred to as a neighbor of the node n_i. The average degree of all the nodes in the network is represented as $1/|N|\sum_{i=1}^{|N|} \aleph_i$.
- **Path:** The path between any two nodes n_i and $n_j \in N$ is represented by $P(n_i, n_j) = <n_i, n_{i+1}, n_{i+2}, ... n_{j-1}, n_j>$, which is a distinct sequence of nodes connected with each other. The cardinality of the path is called the distance. It is possible that there might exist many paths between any two nodes. The path with the **shortest distance** between any two nodes, is a path set containing the minimum number of nodes among all the paths set.
- **Diameter:** It is the cardinality of the longest path among all the shortest paths for all pairs of nodes. Hence, it provides information about the maximum distance among all pairs of nodes.
- **Average Path Length (APL):** It is the average of all the shortest distances for all pairs of nodes represented as $1/|D_{All}|\sum_{i,j\in N\&i\neq j}^{|N|} d(n_i,n_j)$. Where D_{All} represents the set of total shortest paths among all pairs of nodes, and $d(n_i, n_j)$ represents the shortest distance between two nodes n_i, $n_j \in N$. APL provides information about an average on how distant any two nodes are in the network. APL and Diameter can be used to infer how stretched or spread a network is.
- **Edge Density:** Let E_R represent total edges in the network then edge density is defined as the ratio of total edges being present in the network to the total edges that

should be present in an ideal case and can be represented as 2*E_R/N*(N-1), where N is nodes in the network. The number of edges in an ideal case can be N*(N-1)/2 , if all the nodes are connected with each other.

- **Clustering coefficient (CC):** Local CC is defined as the ratio of the number of edges being present among the neighbors of the node under observation to the total number of edges possible among the neighbors. Average CC is calculated by averaging the CC of all the nodes of the network. Edge Density and CC can be used for inferring how tightly knit a network is.
- **Disconnected components:** Some set of nodes might not have paths between them thus, creating disconnected components in the network.

A. Community Detection:

In a complete connected graph, some of the nodes might be more densely connected with each other compared to rest of the nodes in the network. The set of densely connected nodes form a community. The concept of community in a graph is analogous to a set of people in a city which are more closely connected with each other due to cultural and religious backgrounds compared to other inhabitants of a city. In this work, we used Louvain algorithm to detect communities in the network [18]. This greedy optimization method is a widely used algorithm because of its fast execution time in handling very large graphs.

III. Methodology

We investigated the flocking and migration patterns of advocates within and across GitHub and Stack Overflow using social network analysis.

A. Dataset

To understand the communities of advocates and their migrations, we collected data from GitHub and Stack Overflow. Figure 1 shows the overall data extraction process followed to identify the flocks on both sites.

Advocates: To determine advocates (common users across both sites), we selected users who provided GitHub links in their Stack Overflow profiles. 12,578 advocates were found between the two sites using this technique. Unlike past studies [36], we could not use emails to identify common users across GitHub and Stack Overflow because Stack Overflow no longer provides email ids of users in their public database to protect user privacy.

GitHub: We collected GitHub related data using GHTorrent - a public off-line mirror of GitHub data offered using the GitHub REST API [5]. GHTorrent has been used extensively by researchers (such as [36], [32], [16]) and includes a valid dataset of GitHub. The dataset existed in the form of SQL tables on Google BigQuery [11]; hence, we extracted and processed the data for our research questions. Using the 'commits' and 'projects' tables, we filtered out the `user_id` and `commit_SHA` for each common advocate. `commit_SHA` on GitHub is a unique identification key for a commit (an instance of contribution) to a project.

To collect the data related to advocates making changes to a file in a project, we web-crawled GitHub using the package `scrapy` of `Python` language. As GHTorrent did not have data on who edited what file, this was done by generating the links by appending the `commit_SHA` to the url on GitHub's search page and following the web page of that commit. From the commit page, we could pull the list of all files that were affected by that commit. We worked with the GHTorrent data dump of Sept'16.

Stack Overflow: To collect the Stack Overflow related data, we used BigQuery dataset [10], which is updated on a quarterly basis. This dataset has an archive of Stack Overflow content, which included `posts, votes, tags, answers, comments, and badges`. Our dataset was extracted on April'18.

B. Model

We analyzed the network of GitHub and Stack Overflow advocates as a social network and then created a representative model using graph theory. Let $G(N,E,P,F)$ be an undirected graph representing the advocates and their connections in a site, where N represents all the advocates of a production site under investigation, and E represents a set of connections among the advocates. P represents a set of all the projects in the case of GitHub and posts for Stack Overflow. F represents a set of all the files for GitHub and all the interactions for Stack Overflow. Two advocates, N_i and N_j, in G, are connected if they have committed to the same file F in a GitHub project or interacted by answering/commenting to a post in Stack Overflow.

C. Entities

Based on this model, the basic entities are:

For GitHub

- `Project`: Each GitHub project repository represents a separate community network, where one or more advocates might be contributing to the project. Advocates can commit, fork, or pull-request for each project.
- `Files`: A project consist of multiple files. Advocates can add, delete, or modify files collaboratively or individually.

For Stack Overflow

- `Post`: Each Stack Overflow post represents a separate community network of one or more advocates. Advocates generally ask, answer, or comment on the posts.
- `Interactions`: An interaction can be all the ways advocates may interact with other advocates such as to ask questions, provide answers, comment, mark a post favorite, or cast votes.

D. Detecting Flocks

We detected flocks using social network analysis. Within this section, to make use of social network terminology, we refer to flocks as communities. Based on the list of advocates,

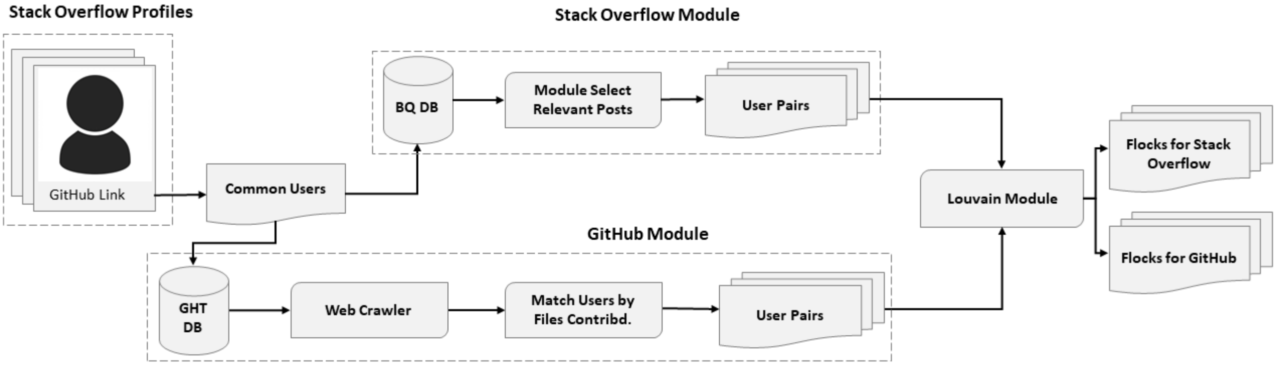


Fig. 1. Data Extraction Process

we detected the communities of these advocates within and across GitHub and Stack Overflow as follows:

Approach for GitHub: We defined connections (E) as 'editing a common file' because, configuration management systems allow advocates to check-out files for making changes, which facilitates rapid parallel software development. Hence, two advocates working on either the same file or interrelated files need to communicate with each other to avoid direct or indirect conflicts [47], [20], [51], [15]. With the list of all files collected from the web crawler, we created a dictionary of each advocate to the files that they worked on. We exported them as `csv` files, each file separated by `project_ID`. Each file thus pertained to one project and held data in the form: {`advocate: [file1, file2, file3, ...]`}. We then created a set of pairwise advocate files for each project. This time, the `csv` was populated with pairs of advocates, like `advocate_1, advocate_2`. A pair would be made if two advocates had worked on the same file in the same project, and this process was accomplished by `Python` script. After determining the pairs of advocates who contributed to the same file in each project, we collected 860 pairs of advocates across all the project files. Then we generated *csv* files of each project. During this process we removed some of the advocates as either they had not contributed or related projects were not publicly accessible. Hence, 951 advocates were left.

We analyzed each of the projects using `iGraph Louvain` module to generate communities. The `Louvain` algorithm discovers communities within networks [18]. It sorts communities based on edge/interaction density and uses a greedy heuristic algorithm to efficiently run in $O(nlogn)$. We used the `Louvain` algorithm implementation in the `Python-iGraph` [3] package in `Python`. Each project ran through the `Louvain` algorithm produced an output file of clusters, or communities, of advocates. These clusters were mutually exclusive in regard to their member advocates, so no advocate appeared in two clusters for a single project.

Approach for Stack Overflow: In the case of Stack Overflow, we used `BigQuery` [10] and `BigQuery_Helper` [4] package to generate our dataset. We used *`Python`* `2.7` along with `pandas` [9], `numpy` [7], `os` [8], and `itertools` [6] for data processing and `iGraph` [3] for generating the communities. To collect our required dataset, we used *posts_questions*, *posts_answers*, and *comments* tables from Stack Overflow database. We generated a list of 9445 `csv` files for each post, which consisted of all the interactions between advocates. We mapped `id` from the *posts_questions* table with the `parent_ids` of *posts_answers* table and `posts_ids` of *comments* table for all advocates.

Finally, the collected data was organized into a `pandas` data frame, with 3 columns - `user_ids, posts_ids, ids_interacted`. This file indicates all the possible interactions within a post in the form of answers or comments. We also found some users commented and answered on their own posts, we filtered those out, since this type of interaction cannot form a community (two or more advocates who interact). Hence, 1104 advocates were left.

The list of `csv` files for each post were then parsed to the `iGraph Louvain` module to generate communities. To achieve this, we used *Graph.Read.Ncol()* to generate the graph objects for each file. These graph objects were then used to generate communities for each `csv` file using `community_multilevel()`. We only took those communities which had two or more advocates in it and found 1250 posts.

E. Detecting Migration

We detected migration both within and across GitHub and Stack Overflow by noting flock movement (migration) in projects/posts. For instance, we created subsets from the cluster/community files generated by the `Louvain` algorithm as follows: (`advocate_1`, `advocate_2`, `advocate_3`) will have subsets (`advocate_1`, `advocate_2`), (`advocate_2`, `advocate_3`), and (`advocate_1`, `advocate_3`) respectively. For each subset, we checked for a match among communities by using the `issubset()` function of `Python` and counted the migrations (number of matches). Hence, the subsets of advocates that matched helped to extract communities that migrated within and across the websites.

F. Characteristics of Advocates

We analyzed the following characteristics.

For GitHub

- Language of Expertise
- Project Owned
- Location

For Stack Overflow

- Language of Expertise
- Reputation
- Location

`Language of Expertise` suggests the advocates' propensity towards the type of programming language they are an expert on since it can be categorized as both a social and technical skill [49].

`Projects Owned` and `Reputation` are indicators of an advocate's expertise. Projects Owned can be categorized as a technical skill and Reputation as a social skill [49]. We calculated the Projects Owned and Reputation Score for each community by

$$abs(diff(R_i,R_{i+1},...,R_n)) \quad \forall i,$$

where i is the number of advocates and R_i is the Projects Owned or Reputation Score of the i^{th} advocate. The threshold θ is calculated according to the following equation:

$$mean((abs(diff(R_i,R_{i+1},...,R_n)_j)) \quad \forall i,j,$$

where i is the number of advocates and j is the number of communities. The notion is, the lower the value of Projects Owned or Reputation Score means the closer the advocates are in terms of strata.

`Location` presents insights about whether advocates prefer to collaborate based on geographic locations - as Lima et al. [37] mentioned users tend to interact with people that are close, as long-range links have a higher cost. We found 301 advocates out of 951 in GitHub and 166 advocates out of 1104 in Stack Overflow who did not have a Location attribute. Since the percentage of missing data is 31% (for GitHub), and approximately 15% (for Stack Overflow), we still reported for this attribute, as past research [37] found that close geographic proximities lead to more collaborations on GitHub.

IV. Results

To understand the flocking and migration behavior of advocates within and across GitHub and Stack Overflow, we used macroscopic and microscopic analysis on the collected data set.

A. Macroscopic Analysis

We collected all the nodes (advocates) across all the different projects/posts to understand the macroscopic view of the GitHub and Stack Overflow network (refer Figure 2). Table I provides information about various metrics of these networks. The total nodes in GitHub were less than Stack Overflow. The network of GitHub was more spread out (higher Diameter and Average Path Length) and less dense (lower Density) compared to Stack Overflow. However, in GitHub, the friends of a friend property is more visible (higher Clustering Coefficient) when compared with Stack Overflow. Among both the networks, Stack Overflow was less disconnected.

TABLE I
Macroscopic Analysis and Comparison for Both Sites in Graph Theory

Metrics	GitHub	Stack Overflow
# Nodes	951	1104
# Edges	793	1173
Average Degree	0.8338	1.0625
Average Path Length	7.445656	6.353188
Diameter	18	16
Density	0.001755493	0.001926564
Clustering coefficient	0.279148	0.0227758
# disconnected components	281	123

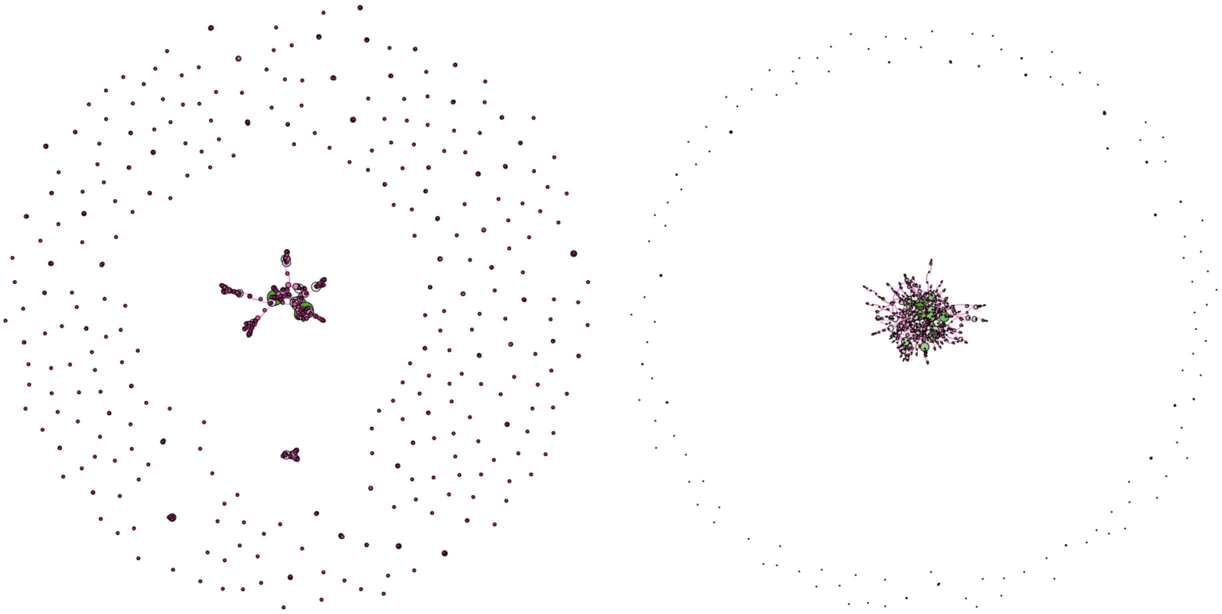
Based on our macroscopic analysis, we observed:

- **Observation 1 – Inter-component interactions:** Figure 2(a) (for GitHub) and Figure 2(b) (for Stack Overflow) give a sense of the disconnectedness in both of these networks. In both the networks, there is a one big connected component (in the middle of the graphs) consisting of 235 (24.7%) and 821 (74.3%) nodes for GitHub and Stack Overflow respectively. Also, the number of disconnected components in GitHub (281) is much more than Stack Overflow (123) compared to the total nodes present in each of the networks. This information can be used to infer that advocates in GitHub are more isolated in general. In addition, low average degree in GitHub also supports this assumption that on average a node interacts with just one other node.
- **Observation 2 – Intra-component interactions:** The high value of the Clustering Coefficient for GitHub can be used to infer that within a disconnected component the advocates work closely together. This is in contrast to inter-component interaction, which appears weak.
- **Observation 3 – Interaction Patterns:** A closer look at the biggest connected components for GitHub (Figure 3(a)) and Stack Overflow (Figure 3(b)) show that in GitHub there are very few nodes that have a high degree of interactions (node size is proportional to the degree/interactions). In other words, most advocates tend to interact with few developers.
- **Observation 4 – Reaching out to the fellow advocates:** Relative high values of diameter and average path length in GitHub indicate that spreading a message to fellow advocates will take more time in GitHub than in Stack Overflow.

> We found that GitHub is not only more disconnected but also has a smaller degree of interactions compared to Stack Overflow.

B. Microscopic Analysis

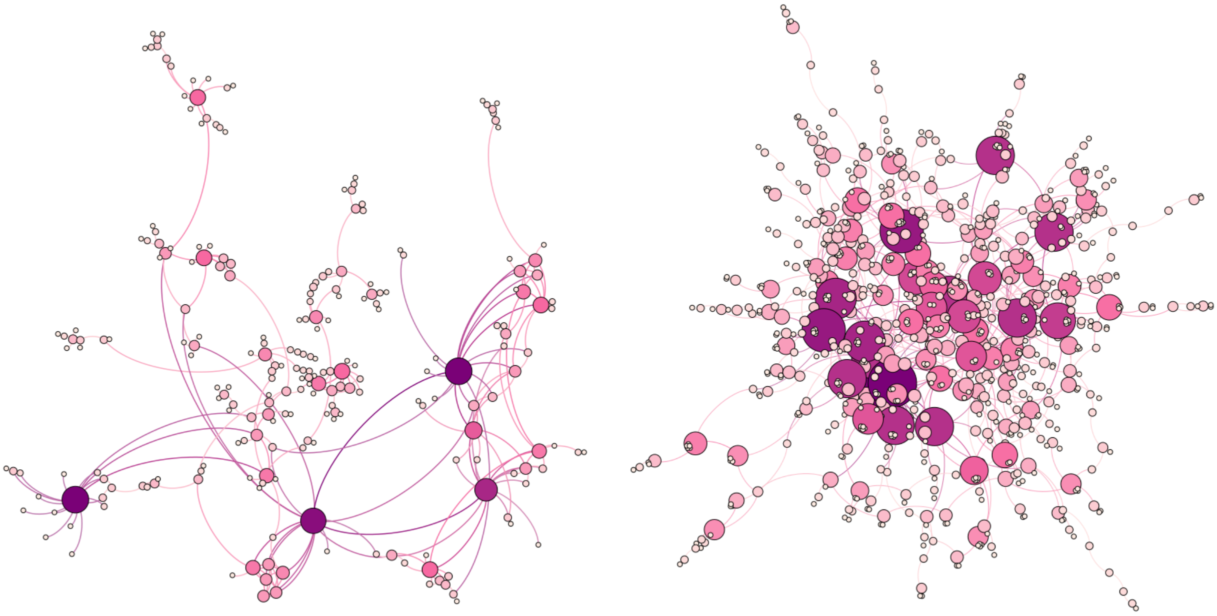
In the macroscopic analysis, we could only observe the flocking behavior of the advocates within GitHub and Stack



(a) GitHub Network

(b) Stack Overflow Network

Fig. 2. Macroscopic View of GitHub and Stack Overflow



(a) GitHub Component

(b) Stack Overflow Component

Fig. 3. Biggest Connected Components of GitHub and Stack Overflow

TABLE II
OVERALL FLOCKING AND MIGRATION PATTERN FOR ADVOCATES WITHIN GITHUB

	Flocks						
# Advocates	2	3	4	5	6	7	11
# of Flocks Found	707	57	8	5	1	1	1
# of Flocks Migrated	109	3	0	0	0	0	0
# of times Flocks Migrated	609	3	0	0	0	0	0

Overflow. Therefore, we performed a microscopic analysis to further address the three research questions.

1) RQ1: Do advocates flock on a peer production site?: To understand the flocking behavior of the advocates, we identified flocks using the `Louvain` algorithm.

(a) Flocks on GitHub:

We found that approximately 7.5% (951/12,578) of advocates flocked within GitHub. In total, 780 flocks were formed by these advocates, with sizes ranging from 2 to 11 advocates in each flock (Refer to Table II).

How do the advocates tend to flock?

We observed 951 advocates formed different sizes of flocks. Table III displays the number of advocates found in different flocks within GitHub. We found that 852 advocates belonged to 1 flock, 132 advocates were part of 2 flocks, and just 1 advocate belonged to 9 flocks. Thus, it's evident from Table III that the general proclivity of advocates is to disperse into congenial groups and form single flocks among themselves than joining multiple flocks.

What characteristics motivated advocates to flock?

We next analyzed all 780 flocks to understand how the characteristics (refer Section III-F) of the advocates played a role in forming the flocks. Table IV shows the characteristics that led to the creation of flocks within GitHub. All 780 flocks' respective advocates had one or more common Language of Expertise. For the Projects Owned characteristic, we found only 71.4% flocks were below the threshold θ (refer Section III-F). Finally, we found 356 (59%) flocks had advocates belonging either from the same *country* or *continent* (*Only 77.05% (601/780) flocks contained location information). Among these flocks, 96 had advocates from the same *continent* and 260 from the same *country*. These results suggest that advocates having the same field of interests and skills form communities (birds of a feather flock together). Hence, people knowing the same languages and living in close proximity tend to create flocks.

(b) Flocks on Stack Overflow:

In Stack Overflow, we found 8.7% (1104/12,578 advocates) flocked within Stack Overflow. These advocates were observed to form 1250 flocks of varied sizes. As seen in Table VI, 1229 flocks were formed with 2 advocates and only 21 flocks were formed with 3 advocates.

How do the advocates tend to flock?

As seen in Table V, we found that 662 advocates belonged to 1 flock, 172 advocates were a part of 2 flocks, 5 advocates belonged to 11 flocks, and 13 advocates were part of 15 or more flocks. Thus, the general trend in forming flocks within Stack Overflow is similar to that of GitHub.

What characteristics motivated advocates to flock?

We analyzed the characteristics that led to the creation of flocks within Stack Overflow (refer Table VII). We used tags from *posts* tables to extract out topics on which a particular advocate preferred to answer or ask a question. We had 1250 flocks of varied sizes, out of which we found the 1024 (81.92%) flocks that stayed together due to one or more interest in topic. For the Reputation criteria, we found 1057 (84.50%) of such flocks that were below the threshold θ (refer Section III-F). Finally, we found 354 (37.98%) flocks had advocates belonging either from the same *country* or *continent* (*Only 74.56% (932/1250) flocks contained location information). Among these flocks, 221 flocks that had advocates from same continent and 133 from same country. Similar to GitHub, we found that advocates with the same interests created flocks in Stack Overflow.

> We found a small percentage–7.5% on GitHub and 8.7% on Stack Overflow– of advocates form flocks within the two sites. Further, we found that advocates with the same fields of interest flocked together on GitHub and Stack Overflow.

2) RQ2: Do the flocks of advocates migrate within a peer production site?

: We investigated the migration pattern of flocks within GitHub and Stack Overflow, i.e., flock of advocates moving from one project/post to another.

(a) Migration on GitHub

Table II summarizes the number of flocks that migrated and number of times these flocks migrated across different projects. We found that 109 flocks with two advocates migrated 609 times over different projects and 3 flocks with 3 advocates migrated 3 times. The flocks with 4 or more advocates did not migrate.

We investigated the total number of flocks (communities) that appeared in multiple projects. Figure 4 shows that the number of flocks contributing to different projects significantly decreased. For example, within GitHub, 59 flocks contributed to 2 projects, 14 contributed to 3 projects, and only one contributed to 23 projects. These results indicate that flocks often were not active in 4 or more different projects on GitHub.

How do the advocates migrate on a code hosting site?

From the 951 advocates that formed different sizes of flocks on GitHub, a total of 171 advocates migrated across different flocks. We investigated the number of flocks an advocate migrate with (refer Table III). 165 advocates migrated with a single unique flock and only 6 advocates migrated with 2 different flocks.

What characteristics motivate a flock of advocates to migrate?

Table VIII summarizes the characteristics of flocks of advocates. For Language of Expertise, all 112 flocks had advocates contribute on code with the same programming language. As for the Project Owned, we found 68.75% (77/112) flocks that

TABLE III
MIGRATION OF ADVOCATES ACROSS FLOCKS WITHIN GITHUB.

Number of	Flocks									Migration	
Flocks involved	1	2	3	4	5	6	7	8	9	1	2
Advocates involved	852	132	37	9	7	2	2	1	1	165	6

TABLE IV
CHARACTERISTICS THAT LED TO FLOCKING IN GITHUB

	Flocks (Found/Total)
Language of Expertise	780/780
Project Owned	557/780
Location*	356/601

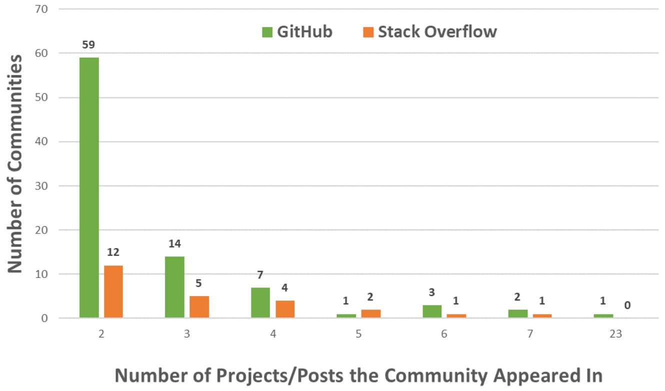


Fig. 4. Number of Flocks vs Number of Projects/Posts Appeared in

had difference lower than θ. Only 88 flocks of advocates revealed their location information. We found 61.36% (54/88) flocks had advocates who belonged to either the same country or continent. Thus, it can be concluded that advocates belonging to the same field of expertise migrate together.

(b) Migration on Stack Overflow

We observed 57 flocks of two advocates migrated 125 times across different posts (refer Table VI). However, no flock of 3 or more advocates migrated.

Figure 4 shows the flocks that appeared across different posts. We recorded 12 flocks that showed up in 2 posts, 5 flocks in 3 posts, and just 1 flock that appeared in 7 posts. These results indicate that flocks were not active in different posts on Stack Overflow.

How does advocates migrate on a Q&A site?

On Stack Overflow, 1,104 advocates formed different sizes of flocks; however, only 256 of them migrated. In Table V, no advocate who was part of a single flock migrated. However, advocates who were involved in more than 1 flock migrated. The instances of such advocates were few and scattered across different sized flocks hence, we decided not to report them. This finding is in direct contrast with the advocates migration pattern which we observed in GitHub. Thus, in Stack Overflow advocates do like to form flocks with like-minded people, but when it comes to migration, they might not stay in the same single flock over time.

What characteristics motivate a flock of advocates to migrate?

Table IX summarizes the three characteristics of flocks for migrating between posts. For Language of Expertise, we found 89.4% (51/57) of flocks that migrated had advocates who worked on the same language as depicted in. For Reputation, 66.66% (38/57) of flocks that had a difference in reputation below the threshold (θ) migrated. In the case of Location, 13 flocks did not have location information. We found 54.5% (24/44) of flocks who had advocates from the same continent or country tended to stick together as they migrate. Hence, the migration in Stack Overflow can be attributed among advocates with same language and reputation but geographically distributed.

> Our results indicate that on GitHub, two advocates who flock based on the same interest migrated across projects. However, on Stack Overflow, a pair of advocates might break their bonds. Further, we also discovered that advocates may learn/help beyond geographic locations while migrating on the GitHub or Stack Overflow.

3) RQ3: Do the flocks of advocates migrate beyond a single peer production site?

: We wanted to see if the flocks of advocates migrated across code hosting (Github) and Q&A sites (Stack Overflow). We found only three flocks migrated across GitHub (containing 1250 flocks) and Stack Overflow (780 flocks). Further, only six advocates migrated across GitHub (951 advocates) and Stack Overflow (1104 advocates). We found these six advocates paired among themselves, forming the three flocks. Further, we observed that these flocks never migrated within GitHub or Stack Overflow. Two out of the three flocks were subsets of larger flocks consisting of three advocates.

What characteristics motivate flocks to migrate across sites?

We also extracted out the characteristics of the advocates that migrated across Stack Overflow and GitHub. For the first flock, both the advocates belonged to the same country and had a reputation of 34,900. However, both these advocates had just one field of interest (Language of Expertise) in common. However, the second flock had a strong relation between their field of interest matching up-to 3 tags and a difference in reputation score of 43,863. However, their location was vastly different, each belonged to a different continent. Finally, for the third flock, both advocates belonged to the same continent, but had a difference in reputation score of 1524 with only one field of interest in common.

TABLE V
MIGRATION OF ADVOCATES ACROSS FLOCKS WITHIN STACK OVERFLOW.

Number of	Flocks													Migration	
Flocks involved	1	2	3	4	5	6	7	8	9	10/11/12	13	14	15 or More	1	2 or More
Advocates involved	662	172	92	59	27	19	13	15	6	5	4	2	13	0	256

TABLE VI
OVERALL FLOCKING AND MIGRATION PATTERN FOR ADVOCATES WITHIN STACK OVERFLOW

	Flocks	
# Advocates	2	3
# of Flocks Found	1229	21
# of Flocks Migrated	57	0
# of times Flocks Migrated	125	0

TABLE VII
CHARACTERISTICS THAT LED TO FLOCKING IN STACK OVERFLOW

	Flocks (Found/Total)
Language of Expertise	1024/1250
Reputation	1057/1250
Location*	354/932

Only three flocks migrated across GitHub and Stack Overflow. The advocates in these three flocks had one or more common characteristics.

V. DISCUSSIONS

Based on our findings, we could predict how future advocates behave in GitHub and Stack Overflow: the advocates in GitHub tended to work within small flocks with selective members who may have the same interests or professionalism in the project, and once they decide to migrate (move) to the next project with their current flock, the prospect of the flock working together on future projects may decrease. In Stack Overflow, however, advocates are not willing to migrate as they do not answer multiple questions asked by the same advocate. In summary, for both sites, we could consider that, it is "unusual" for advocates to collaborate again.

A. Implications

Understanding the flocking and migration behavior of advocates in peer production sites such as GitHub and Stack Overflow can help develop supplemental tools that can promote effective and efficient collaboration between the advocates of those sites. Our findings have a number of implications for tool builders to facilitate flocking and migration behavior within and across peer production sites.

Searching Code based on Social Interactions: The understanding of flock formation can help in the design of code-searching tools based on developers' social interactions. This new paradigm can help in searching for trusted code examples in one's own social network. For example, these tools could leverage socio-technical skills from peer production sites to advertise, monitor, and assess the quality of code contributions using psycho-physiological measures to evaluate task difficulty or manage interruptions.

TABLE VIII
CHARACTERISTICS THAT LED TO MIGRATE FOR GITHUB

	Flocks (Found/Total)
Language of Expertise	112/112
Project Owned	77/112
Location*	54/88

TABLE IX
CHARACTERISTICS THAT LED TO MIGRATE FOR STACK OVERFLOW

	Flocks (Found/Total)
Language of Expertise	51/57
Reputation	38/57
Location*	24/44

Migration within a code hosting site: Our results suggest that flocks formed in GitHub tend to be small, but are restricted more to coordination among individuals and technology specific to their projects. Hence, currently these flocks exist in isolation. Such behavior was also observed by Datta et al. [27]. This suggests we need to study ways to motivate and design tools to support migration within a code hosting site; for example, a tool that identifies similar interests in developers and recommends potential projects or flocks accordingly. This could help improve both quantity and quality of projects in GitHub, since developers could narrow down their range of project selection by running such a tool to save time and enhance the quality of projects due to similar internal motivation interest of developers.

Migration across peer production sites: Our results suggests little migration of flocks across the peer production sites. Hence, there is a need to build predictive analytic and recommendation applications for supporting migration across peer production sites. For instance, prediction based models that observe the socio-technical activities of advocates on multiple peer production sites could recommend advocates for collaboration. The models could recommend flocks across communities on both code hosting and Q&A sites.

Recommending flocks to newcomers: Based on the communication and incentive structures of peer production sites, a recommendation system can be developed for newcomers to find mentors among advocates and help them in forming flocks for future collaborations.

VI. THREATS TO VALIDITY

Bias due to sampling and dataset: We collected advocates from a single project-hosting website, GitHub, and a single question and answer website, Stack Overflow. Thus, our conclusions may not be perfectly generalizable. However, these are the most popular peer production sites, and they represent the vast majority of developers for creating open source software.

Secondly, our dataset of advocates is 12.5K, which is much smaller than the past research [16], as they collected 92K common developers using MD5 hashes, which are no longer accessible due to privacy reasons. Hence, our data is not representative of the total number of advocates. We argue that although our data is small, it is accurate and precise as we used GitHub links from Stack Overflow profiles to collect advocates.

Bias due to used metrics: We defined communities as groups of advocates who contributed to the same file in a project which is one of the many possible ways to define and detect communities. With this metric, we did not consider time factor while creating communities. Hence, a threat can arise given that two developers who worked on the same file years apart probably should not be considered a community. We argue that this scenario can not occur in our analysis as we filtered out any communities that only appear for one file. In Stack Overflow, the data collected from BigQuery returned data tuples with comments associated with posts, which are not representative of real scenarios where comments are associated with either a specific question or answer of a post. Hence, the data collected for a post may not be representative of a real scenario but it does represent the overall community for a post.

Further, we selected few characteristics for GitHub and Stack Overflow to do in-depth analysis. Other characteristics such as age, up-votes, down-votes, number of files modified, type of project, etc. were not considered. We decided to focus on the few characteristics as they were present across the two sites and could help with predictions on flocking and migration behaviors on these sites.

Bias due to community detection algorithm: The selection of Louvain community detection algorithm is based on the fact that it is one of the most popular and efficient algorithms. However, it only returns communities which are either very small or large. Thus, this may affect the community analysis in general.

VII. Related Work

A. Social Networks

In the last two decades, researchers have investigated different social entities in various domains to determine if the concept of flocking behavior is prevalent or not. For example, in politics, researchers investigated the interactions among politicians and citizens on Twitter, specifically those from North American and European countries, and found compelling evidence of homophily (flocks) [35], [14], [17], [33]. Tang et al. [54] conducted a longitudinal study on scientific collaborations and found a strong flocking behavior. In another study, researchers investigated ethnic and cultural roles in the creation of friendship (flocks) among adolescents [57]. The flocking behavior is also observed among spammers, who end up spamming the same users based on their common intentions [22].

Migration behavior due to homophily has been explored in various domains. In particular with respect to immigrants [53], researchers have studied health among minority members [46]. Lu et al. [39] studied mobile data to understand migration patterns of a community after a natural disaster and found that people migrate to the places they were making frequent phone calls to before the disaster.

Even though our study is related to flocking and migration behaviors in online peer production sites, it cannot be isolated from the basis of a social network since each site is considered as a community of different sizes. Hence, we utilize the theories of social network analysis to understand the human (advocates) behavior in these virtual communities.

B. Software Engineering

GitHub and Stack Overflow have been two popular online peer production sites for programmers for the past ten years. GitHub has been the largest open source site for code hosting and version control. It has more than 2.1B businesses and organizations, 40M developers worldwide, and 100M repositories [1]. Researchers have concluded that the interactions on GitHub can be viewed as social activities [55], and developers' behavior is largely influenced by the awareness of the fact that they are being observed by their peers [25]. Stack Overflow is currently the largest online community for developers to build their careers by learning and sharing programming knowledge via Question/Answer. It contains over 19M questions and 29M answers, 50M visitors each month, and has helped developers over 43.3B times [2]. Mamykina et al. [41] showed that most Stack Overflow questions are answered in a median time of 11 minutes, providing quick solutions to technical problems. Research has explored how Stack Overflow encourages participants to ask "good" questions and to give "good" answers through reputation incentives, such as points and badges [21].

Past research has studied community formation, or flocks, extensively on GitHub. Dabbish et al. [25] highlights many of the motivations behind users forming communities, including the facilitation of communication, streamlining of technical goals, collective inference of project outcomes, advancement of technical skills, and management of reputation. Lima et al. [37] discussed social interactions on GitHub and found that active users may not necessarily have a large follower base and the users in close proximity according to geographic location are more ready to interact with each other. Thung et al. [55] also uncovered and analyzed inter-project and inter-developer relations on GitHub. Majumder et al. [40] researched GitHub to discover optimal team-formation techniques and algorithms. Yu et al. [61] looked at the power of social programming, linking programming social networks to attracting external developers and causing explosive growth in development. Tsay et al. [56] saw how social connections and interactions influenced accepted pull requests and ultimately the development path of a project. Jiang et al. [34] researched the unfollowing behavior of users on GitHub, providing insights into the relationships between developers and their followers. Brisson et al. [19] analyzed the communication within "software families" (repositories and their forks) and determined how that communication is related to the number of stars on GitHub.

Our paper built upon similar underlying concepts of social interaction on GitHub but extended them to and focused on the homophily flocking and migration of advocates.

The only research focused on peer parity (flocking) within Stack Overflow is from Ford et al. [30], [29]. They conducted a thorough research on peer parity with women in Stack Overflow. They analyzed how women interacted, formed communities, and helped each other within the male-dominated field of software engineering.

Researchers have also performed cross-site studies of users within GitHub and Stack Overflow. Vasilescu et al. [58] discovered that a user's activity and participation on Stack Overflow correlates with their coding activity on GitHub. They specifically observed the correlation between the number of questions asked and answered on Stack Overflow and the number of commits on GitHub by individuals. Badashian et al.[16] performed an in-depth analysis and followed inter-network activity over a five-year period to look for patterns between activity of individual developers on the two websites. Their results showed moderate to strong correlations between each site.

Our work is different from past research as we study the advocates - developers who are active in both GitHub and Stack Overflow, using social network analysis to understand the flocking and migration patterns of advocates within and across both the sites.

VIII. Conclusions

In this paper, we analyzed the flocking and migration behavior of 12.5K advocates – developers who were active on both GitHub and Stack Overflow. Our results from macroscopic and microscopic analysis verify that advocates do flock and migrate to an extent. We found that 7.5% of the advocates create flocks on GitHub and 8.7% on Stack Overflow. Further, these flocks of advocates migrate on an average of 5 times on GitHub and 2 times on Stack Overflow. The results reveal a general trend that advocates in GitHub and Stack Overflow tend to work in small flocks, this pattern may induce less flocking and migration behavior among the advocates. Further, advocates in GitHub are bound by long-term project interactions, which may lead them to be more selective of their collaborators. While, Stack Overflow's interactions are sporadic and short-term, resulting in the creation of more connections. Our findings has implications for software practitioners, researchers, and tool builders to study and support the flocking and migration of advocates in and across different peer production sites.

Acknowledgments

This work is supported by H2020 framework project, SoBigData, grant no. 654024.

References

[1] GitHub Website Stats. http://www.github.com

[2] Stack Overflow Website Stats. https://stackoverflow.com/company

[3] igraph (2014). URL http://igraph.org/python/

[4] BigQuery Helper (2018). URL https://github.com/SohierDane/BigQuery_Helper

[5] Github rest api (2018). URL https://developer.github.com/v3/. Accessed: 2020-04-13

[6] itertools (2018). URL https://docs.python.org/2.7/library/itertools.html

[7] Numpy (2018). URL https://docs.scipy.org/doc/numpy/reference/

[8] os (2018). URL https://docs.python.org/2.7/library/os.html

[9] pandas (2018). URL https://pandas.pydata.org/

[10] BigQuery Stack Overflow Data Set (2020). URL https://console.cloud.google.com/bigquery?p=bigquery-public-data&d=stackoverflow&project=alert-ground-192618. Accessed: 2020-04-13

[11] GHTorrent Data Set (2020). URL https://console.cloud.google.com/bigquery?GK=ghtorrent-bq&page=dataset&d=ght&p=ghtorrent-bq&redirect_from_classic=true&pli=1&proje ground-192618. Accessed: 2018-04-10

[12] Al-Ani, B., Bietz, M.J., Wang, Y., Trainer, E., Koehne, B., Marczak, S., Redmiles, D., Prikladnicki, R.: Globally distributed system developers: their trust expectations and processes. In: Proceedings of the 2013 conference on Computer supported cooperative work, pp. 563–574. ACM (2013)

[13] Al-Ani, B., Redmiles, D.: In strangers we trust? findings of an empirical study of distributed teams. In: 2009 Fourth IEEE International Conference on Global Software Engineering, pp. 121–130 (2009). DOI 10.1109/ICGSE.2009.20

[14] Anatoliy, G., Jeffrey, R.: Investigating political polarization on twitter: A canadian perspective. Policy & Internet **6**(1), 28–45 (2014)

[15] Arciniegas-Mendez, M., Zagalsky, A., Storey, M.A., Hadwin, A.F.: Using the model of regulation to understand software development collaboration practices and tool support. In: CSCW, pp. 1049–1065 (2017)

[16] Badashian, A.S., Esteki, A., Gholipour, A., Hindle, A., Stroulia, E.: Involvement, contribution and influence in github and stack overflow. In: CSCW, pp. 19–33. IBM Corp. (2014)

[17] Barbera, P.: Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data **23** (2013)

[18] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks (2008)

[19] Brisson, S., Noei, E., Lyons, K.: We are family: Analyzing communication in github software repositories and their forks. In: 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 59–69 (2020)

[20] Brun, Y., Holmes, R., Ernst, M.D., Notkin, D.: Proactive detection of collaboration conflicts. In: FSE, pp. 168–178 (2011)

[21] Capiluppi, A., Serebrenik, A., Singer, L.: Assessing technical candidates on the social web. IEEE Software **30**(1), 45–51 (2013)

[22] Cohen, Y., Hendler, D.: Birds of a feather flock together: The accidental communities of spammers. In: IEEE/ACM ASONAM, pp. 986–993 (2015)

[23] Cook, J., Smith, M.: Beyond formal learning: Informal community elearning. Computers & Education **43**(1), 35–47 (2004). URL http://www.sciencedirect.com/science/article/pii/S0360131503001416

[24] D., F.N., Pascale, Q.: Birds of a feather flock together ... definition, role and measure of congruence: An application to sponsorship. Psychology & Marketing **24**(11), 975–1000 (2007)

[25] Dabbish, L., Stuart, C., Tsay, J., Herbsleb, J.: Social coding in github: Transparency and collaboration in an open software repository. In: CSW12, pp. 1277–1286. ACM (2012)

[26] Dabbish, L.A., Stuart, H.C., Tsay, J., Herbsleb, J.D.: Social coding in github: transparency and collaboration in an open software repository. In: CSCW (2012)

[27] Datta, S., Bhatt, D., Jain, M., Sarkar, P., Sarkar, S.: The importance of being isolated: An empirical study on chromium reviews. In: ACM/IEEE ESEM, pp. 1–4 (2015)

[28] Deterding, S.: Gamification: designing for motivation. interactions **19**(4), 14–17 (2012)

[29] Ford, D.: Using eye tracking to identify features of peer parity on stack overflow. In: VL/HCC, pp. 319–320 (2017)

[30] Ford, D., Harkins, A., Parnin, C.: Someone like me: How does peer parity influence participation of women on stack overflow? In: VL/HCC, pp. 239–243 (2017)

[31] Gershenson, S., Hart, C.M.D., Lindsay, C.A., Papageorge, N.W.: The Long-Run Impacts of Same-Race Teachers (2017)

[32] Gousios, G.: The GHTorrent dataset and tool suite. In: MSR, pp. 233–236 (2013)

[33] Itai, H., Stephen, M., Marc, S.: Birds of a feather tweet together: Integrating network and content analyses to examine cross-ideology exposure on twitter. Journal of Computer-Mediated Communication **18**(2), 154–174 (2013)

[34] Jiang, J., Lo, D., Yang, Y., Li, J., Zhang, L.: A first look at unfollowing behavior on github. Information and Software Technology **105**, 150 – 160 (2019)

[35] Larsson, A.O., Ihlen, A.: Birds of a feather flock together? party leaders on twitter during the 2013 norwegian elections. European Journal of Communication **30**(6), 666–681 (2015)

[36] Lee, R.K.W., Lo, D.: Github and stack overflow: Analyzing developer interests across multiple social collaborative platforms. In: SocInfo (2017)

[37] de Lima, A.M.G., Rossi, L., Musolesi, M.: Coding together at scale: Github as a collaborative social network. CoRR (2014)

[38] Long, J.: Open source software development experiences on the students' resumes: Do they count?-insights from the employers' perspectives. Journal of Information Technology Education: Research **8**(1), 229–242 (2009)

[39] Lu, X., Bengtsson, L., Holme, P.: Predictability of population displacement after the 2010 haiti earthquake. pp. 11,576–11,581. National Academy of Sciences (2012)

[40] Majumder, A., Datta, S., Naidu, K.: Capacitated team formation problem on social networks. In: KDD, pp. 1005–1013 (2012)

[41] Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., Hartmann, B.: Design lessons from the fastest q&a site in the west. In: CHI, pp. 2857–2866. ACM (2011)

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

[43] Marlow, J., Dabbish, L., Herbsleb, J.: Impression formation in online peer production: activity traces and personal profiles in github. In: Proceedings of the 2013 conference on Computer supported cooperative work, pp. 117–128. ACM (2013)

[44] Movshovitz-Attias, D., Movshovitz-Attias, Y., Steenkiste, P., Faloutsos, C.: Analysis of the reputation system and user contributions on a question answering website: Stackoverflow. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 886–893. ACM (2013)

[45] Parnin, C., Treude, C., Grammel, L., Storey, M.A.: Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. Tech. rep. (2012)

[46] Rostila, M.: Birds of a feather flock together and fall ill? migrant homophily and health in sweden. Sociology of health and illness (32) (2010)

[47] Sarma, A., Bortis, G., van der Hoek, A.: Towards supporting awareness of indirect conflicts across software configuration management workspaces. In: ASE, pp. 94–103. ACM (2007)

[48] Sarma, A., Chen, X., Kuttal, S., Dabbish, L., Wang, Z.: Hiring in the global stage: Profiles of online contributions. In: Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on, pp. 1–10. IEEE (2016)

[49] Sharma, A., Chen, X., Kuttal, S., Dabbish, L., Wang, Z.: Hiring in the global stage: Profiles of online contributions. In: ICGSE (2016). DOI 10.1109/ICGSE.2016.35

[50] Singer, L., Figueira Filho, F., Cleary, B., Treude, C., Storey, M.A., Schneider, K.: Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators. In: Proceedings of the 2013 conference on Computer supported cooperative work, pp. 103–116. ACM (2013)

[51] Steinmacher, I., Chaves, A.P., Gerosa, M.A.: Awareness support in distributed software development: A systematic review and mapping of the literature. CSCW **22**(2-3), 113–158 (2013)

[52] Subramaniam, M.M., Ahn, J., Fleischmann, K.R., Druin, A.: Reimagining the role of school libraries in stem education:creating hybrid spaces for exploration. The Library Quarterly **82**(2), 161–182 (2012)

[53] Szebenyi, D.: https://www.liberties.eu/en/infographics/migration-in-the-eu-infographic/112. Online Accessed: 2016-01-22

[54] Tang, L.: Does "birds of a feather flock together" matter - evidence from a longitudinal study on us-china scientific collaboration. Journal of Informetrics **7**(2), 330 – 344 (2013)

[55] Thung, F., Bissyande, T.F., Lo, D., Jiang, L.: Network structure of social coding in github. In: CSMR, pp. 323–326. IEEE (2013)

[56] Tsay, J., Dabbish, L., Herbsleb, J.: Influence of social and technical factors for evaluating contribution in github. In: ICSE, pp. 356–366 (2014)

[57] V. Hamm, J.: Do birds of a feather flock together? the variable bases for african american, asian american, and european american adolescents' selection of similar friends **36**, 209–19 (2000)

[58] Vasilescu, B., Filkov, V., Serebrenik, A.: Stackoverflow and github: Associations between software development and crowd sourced knowledge. In: 2013 SocialCom, pp. 188–195 (2013)

[59] Wenger, E.C., Snyder, W.M.: Communities of practice: The organizational frontier. Harvard Business Review **78**(1), 139–146 (2000)

[60] Yang, S.H.: Using blogs to enhance critical reflection and community of practice. Journal of Educational Technology & Society **12**(2), 11–21 (2009). URL http://www.jstor.org/stable/jeductechsoci.12.2.11

[61] Yu, Y., Yin, G., Wang, H., Wang, T.: Exploring the patterns of social behavior in github. In: CrowdSoft, pp. 31–36 (2014)