Competence-Confidence Gap: A Threat to Female Developers' Contribution on GitHub

Zhendong Wang
Department of Informatics
University of California, Irvine
Irvine, CA
zhendow@uci.edu

Yi Wang
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY
yi.wang@rit.edu

David Redmiles
Department of Informatics
University of California, Irvine
Irvine, CA
redmiles@ics.uci.edu

ABSTRACT

On GitHub, contributing to a new project is crucial for a developer to gain personal growth and maximize impact in the community. It is known that female developers are often hesitant to explore the opportunities to contribute to new projects even when they possess the competence to make valuable contributions. Drawing from the literature of the competence-confidence gap, we develop a fresh explanation for this phenomenon. We validate the theoretical explanation through an empirical study using GitHub's historical data. In this study, we identify all female developers ranking in top 5,000 GitHub users. Using the Granger Causality Test, we find that, for the majority of identified female developers, initiating a pull request to a new repository is "Granger" caused by the quick increase of followers in the preceding couple of weeks. For most male developers, our observations show that their new pull requests have no relationship with the dynamics of follower numbers. The results indicate that the competence-confidence gap is a threat to female developers' contribution on GitHub. The research suggests that helping female developers to overcome the competence-confidence gap is critical for encouraging female's contribution open source development, as well as growing their reputations and impacts in the community.

CCS CONCEPTS

• **Social and professional topics** → **Women**; • **Human-centered computing** → *Empirical studies in collaborative and social computing*; • **Software and its engineering** → *Open source model*;

KEYWORDS

Female Developers, Competence-Confidence Gap, Granger Causality, GitHub

ACM Reference Format:
Zhendong Wang, Yi Wang, and David Redmiles. 2018. Competence-Confidence Gap: A Threat to Female Developers' Contribution on GitHub. In *Proceedings of 40th International Conference on Software Engineering: Software Track, Gothenburg, Sweden, May 27-June 3, 2018 (ICSE-SEIS'18),* 10 pages. https://doi.org/10.1145/3183428.3183437

ICSE-SEIS'18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5661-9/18/05…$15.00
https://doi.org/10.1145/3183428.3183437

1 INTRODUCTION

Recent research has made it clear that women are underrepresented in the software development industry and suffer from various types of prejudice and bias [43]. According to a recent labor market census, in 2015, female developers only account for 21% of the whole software development workforce and earn $19,514 less than their male peers [14] on average. Many male developers claim that women are less qualified for software development careers. James Damore, a former Google employee, wrote a widely distributed anti-diversity memo which is full of discrimination towards his female colleagues [13]. It is fair to say that there are limited opportunities in today's corporate world for women to pursue a distinct career except for a few well-known exceptions [3].

Since the opportunities for formal employment are limited, many women participate in software development by contributing to open source development, which is assumed to be a meritocracy-based system and free of gender barriers [62]. However, the real situation is not that positive. On GitHub, which is one of the most popular platforms for open source software development, women own only less than 5% of projects [19]. There is no woman in the top 100 lists, measured either by contribution or popularity [44].

GitHub and other open source platforms are indeed socio-technical systems [6]. It is no surprise that such a platform and its users inherit the historical biases and prejudice towards women [60]. While changing the social prejudice and biases is a mission that may take a few generations, an individual female developer can be more proactive and make contributions to open source projects, and grow her impact in the community. However, a regrettable fact is that female developers are often hesitant to make contributions to open source projects using the competitive pull-request model even when they do write better code than males [60]. So, why?

Drawing from the literature on the competence-confidence gap [27], we develop a theoretical explanation for female developers' low rate of initiating a pull request to a new project. We argue that this partially results from the fact that it is not easy for female developers to directly translate competence to confidence. To establish confidence, in this technical sense, they need certain levels of social attraction (being liked by the other community members) [32]. In contrast, male developers can directly translate competence to confidence irrelevant of social attraction. To validate this theoretical explanation, we design a mixed-method empirical study using the historical data from GitHub. We construct a unique dataset consisting of time series data from 168 manually identified top female developers and randomly sampled 168 male developers from the top 5,000 GitHub user list based on followers number. Through applying the Granger causality test, we find support to the competence-confidence gap and the moderating effect of social attraction in translating competence to confidence among female developers. Using the qualitative data of the sampled top female developers, we reveal how they overcome the competence-confidence gap and become influential.

Our work has significant social implications. First, we confirm the existence of the competence-confidence gap, which may help female developers better understand the dynamics at play in GitHub and change any negative self-assessment of confidence and competence. Second, the moderating effect of social attraction allows potential mechanism interventions and user experience designs for encouraging female developers' contributions. Third, while popular media outlets such as Huffington Post [57] provide various high-level general suggestions for women to overcome competence-confidence gap, our work suggests actionable and proven effective strategies specific for open source software development. Lastly, as our findings reveal that female developers are actually being rewarded for respecting the male-dominated social norm on GitHub, it reminds us that there is still a long way ahead to truly achieve gender equality, not only in software development industry but the whole society.

In summary, the contributions of this paper are as follows:

- Developing a theoretical explanation based on competence-confidence gap literature;
- Performing an empirical study validates the theoretical explanation through sophisticated data analysis techniques;
- Discussing the mechanism and user experience design implications for encouraging female developers' contribution to open source projects;
- Introducing a mature, well-accepted technique for establishing and reasoning causality relationship to software engineering community.

The rest of the paper proceeds as follows. Section 2 introduces the competence-confidence gap theory, which is the theoretical foundation of our analyses. Section 3 presents the study design while section 4 describes how we construct the empirical dataset. Section 5 summarizes the methods we used to analyze the data. Section 6 presents the results of data analyses. Section 7 discusses the implication of the research. Section 8 and 9 briefly introduces the related work and concludes the paper respectively.

2 THEORY AND PROPOSITION

2.1 Competence-Confidence Gap

Psychology and education researchers have observed that women are often lacking confidence even though they may already developed competence, identifying the so-called "competence-confidence gap." For example, Pomerantz et al. [50] show that even girls doing well in school suffer a higher level of internal stress from self-doubt than boys do. Particularly in STEM areas, there is a rich body of evidence showing that women do not exhibit the comparable level of confidence with their male peers, e.g., [42, 66].

The competence-confidence gap has little to do with a woman's achievement. Even the most successful women often cannot escape from the trap of competence-confidence gap sometimes. For example, Sheryl Sandberg, COO of Facebook, told The Atlantic in an interview [37] a year before her book, *Lean In*, was published: "*There are still days I wake up feeling like a fraud, not sure I should be where I am.*" However, the picture of competence and confidence is totally different for men. In his article "Why Do So Many Incompetent Men Become Leaders?" in Harvard Business Review, Chamorro-Premuzic wrote: "*because we (people in general) commonly misinterpret displays of confidence as a sign of competence, we are fooled into believing that men are better leaders than women* [10]."

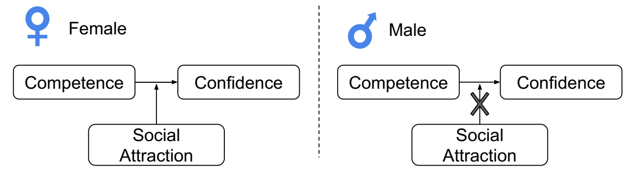

Figure 1: The differences between men and women on how to translate competence to confidence, adapted from [27].

Being confident is important. At the individual level, a person of low confidence may lose many important opportunities in her life such as career development. For example, a confident job interviewee is more likely to receive an offer compared with others who have similar background and skill but less confident [59]. For organizations, management literature has demonstrated confident employees often lead to desirable organization outcomes such as creativity [47], learning orientation [21], and so on. The natural result of low confidence is inaction. The lack of confidence makes women unable to fully leverage their competence to achieve career successes. They are often hesitant to explore new opportunities for their personal growth and career advancement. Even when they are capable, they may still need to spend extra effort to convince herself to take actions [27].

Similar situations exist in software development activities. Based on data from a software development multinational company, Guillén et al. demonstrates the differences between men and women on how an individual's competence "translates" to her/his confidence (Fig. 1 shows the differences of this translation process). Their theory features the moderating role of social attraction. It is defined as the extent to which others like an individual and want to bond with him/her. For women, only possessing competence (e.g., good software development skills) are not enough to make them confident. Feeling to be liked is also important for them to be confident. However, for men, whether or not being liked is irrelevant to the process of translating competence to confidence [27, 57].

2.2 Competence-Confidence Gap in GitHub

As a collaborative software development platform, GitHub is indeed a socio-technical system [6]. Therefore, it inherits the established social norms, values, and even biases existing in our society. Although GitHub is not a formal organization, its users are likely to follow the social norms formed in other social contexts [58]. Moreover, many projects on GitHub, particularly the well-established large ones, adopted the pull request model [36, 61]. The pull request model is a competitive contributing model that a user's contribution is not guaranteed to be accepted [62], hence creates the context making its user feel peer pressure and competitions which the users often experience in their formal organizations.

GitHub implements a set of social mechanisms to encourage their users' continuous participation and contributions [34]. In individual interactions, an important mechanism is the following/follower where a user can follow other users or be followed by other users. The following/follower relationships are an implementation of social bond [40], which would inevitably lead to the social attraction. It is natural for an individual who has a lot of followers to feel that he or she is being liked by other members in the community. Moreover, the evaluations to pull requests, especially the pull request to the project that a user has no interaction history, are not purely based on merit but also relates to the social profiles such as follower numbers [61]. Female developers may have more confidence to initiate a pull request to a new project after a fast growth of follower numbers.

Hence, the context of GitHub, though may be different from formal software development organizations, is analogical to them. Therefore, we believe that the theoretical framework developed in [27] is still applicable. We could expect the existence of competence-confidence gaps among female developers on GitHub while male developers may be free of it. At the observable behavioral level, we can expect that: for female developers, initiating a pull request to a new project happen after a period of quick growth of follower number, while the dynamics of follower number is irrelevant for male developers' decision on trying to contributing to a new project.

2.3 The Main Proposition

Based on above discussions, we formalize the main proposition on the competence-confidence gap in GitHub as follows:

In GitHub, competence-confidence gap is still a widespread phenomenon among female developers. Female developers are often hesitant to make contributions to new projects using competitive contributing model (pull-request model). For them, social attraction moderates the translation from competence to confidence. If they feel that they are being liked in the community (fast increases of followers), they are more likely to attempt to participating a new competitive project by initiating a pull request. For top female developers, we can expect that they exhibit some strategies to overcome the competence-confidence gap. For male developers, they are less likely to be impacted by the competence-confidence gap.

We seek to empirical validate the proposition through mining the GitHub's historical event dataset. In section 3 through 6, we will introduce the empirical study design, dataset, analysis methods, and results. Just provide a brief preview: we do find empirical support to the main proposition.

3 EMPIRICAL STUDY DESIGN

3.1 Theoretical Predictions

The theoretical propositions indicate the following three observable behavioral predictions:

P1. For the majority of female developers on GitHub, their pull requests to new projects often happen after a period of fast increase of followers.

P2. For the majority of female developers on GitHub, their pull requests to new projects have no apparent relationship with the past growth of the number of followers.

To empirical validate the theoretical propositions, we need to find empirical support for the predictions. To do so, we design a empirical study following well-established guidelines in empirical software engineering literature, such as Easterbrook et al. [16], and Kitchenham et al. [39]. The empirical study takes the mixed-method approach with a sequential exploratory strategy. We employ quantitative methods to identify evidence for **P1** and **P2**.

Because the question of "true causality" is deeply cultural and philosophical [54], and the historical arguments between "post hoc ergo propter hoc" and "cum hoc ergo proper hoc", statistical methods, even those contain the word "causality" in their names (e.g., Granger causality in this study) are mostly about "predictive causality" [5, 26]. To demonstrate the "true causality", it has to be combined with the theoretical reasoning, which is exactly what we do in the prior section. Hence, by combining theoretical proposition (reflected by its behavioral predictions) developed in section 2 and the Granger causality test, our study completes a Deductive-Nomological Model of scientific explanation [30] which essentially establishes a true causal inference [48]. Hence, we can demonstrate that the competence-confidence gap is *at least* a factor of female developers' low rate of attempting to contribute to a new project.

3.2 Research Setting

We target GitHub which is one of the most popular collaborative software development platforms. It hosts many important open source projects such as `iPython`, `Ruby on Rails`, and so on. GitHub's pull-based development model is different from traditional OSS. On GitHub, typically only a small group of developers (the core) have access to the main repository and can push their code directly. All other contributors work in isolation in their local copies of the repository (forks), and submit their contributions back for review and integration through pull requests. Therefore, there is no guarantee that a pull request will be accepted and merged to the main repository. There may be multiple competing pull requests for a single work item, hence, the pull request model is a competitive contribution model. It provide an excellent natural setting that makes the "inaction" caused by lack of confidence and "action" (pull request) after gaining confidence observable.

We focus on the female developers among top 5,000 users on GitHub. This is based on three considerations. First, the top 5,000 users only account for 0.02% of GitHub's over 25M users, thus they are real elite users. Studying them does not only enable us to validate the proposition but also helps us identify how top female developers overcome the competence-confidence gap (if it exists). Second, most of them have made a number of nontrivial contributions to multiple repositories under the pull request model. Using pull request data help us to avoid several perils of mining GitHub repositories [36]. Lastly, from the practical perspective, the distribution of follower numbers exhibits the long tail pattern. When an individual's follower number is small relative to her tenure, which

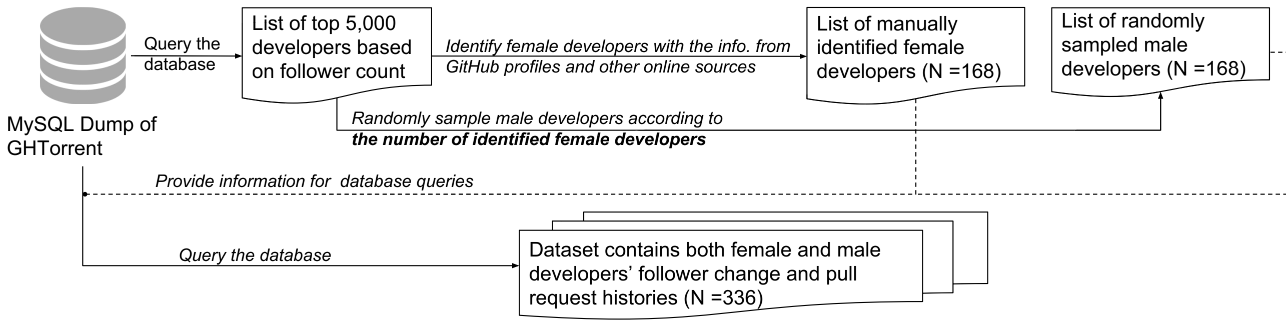

Figure 2: The data extraction process.

is usually 5 to 8 years, random effects may become significant for the dynamics of follower numbers. This might hurt the reliability of the statistical analyses and inferences.

4 DATASET

We construct a unique dataset for the study. The main data source for the study is publicly-available GitHub data from the GHTorrent [23]. The GHTorrent retrieves GitHub public event from its timeline and stores the data at a MongoDB database as well as a MySQL mirror. In this study, we query data from the MySQL dump of GHTorrent on *09-01-2017*. The data dump contains the public events from 2008 until 2017.

4.1 Data Extraction

Fig. 2 depicts the steps of extracting the necessary raw data from the MySQL dump. The MySQL dump of GHTorrent is downloaded to a local Windows server and restore into the local MySQL database. By joining two data tables, users and followers, we extract the unique identification (login), of the top 5,000 developers on GitHub based on their followers count. The login is directly associated with the developer's GitHub homepage by access the URL: *github.com/login*.

4.2 Identifying Top Female Developers

GitHub does not request information about a user's gender identification. Although there are a few approaches to automatically infer a user's gender [63], they are based on data from online communities such as the Stackoverflow [63] other than GitHub and have not been empirically validated with GitHub data[1]. Terrell et al. suggest a method that using email to link a user to her Linkedin profiles [60]. Although this method is high on precision, its performance is not very good on recall. Particularly for a large proportion of top female developers from East Asian countries where Linkedin's penetration is not that high, this method can only recognize a relatively small proportion of female developers in the top 5,000 list.

[1]Vasilescu et al. (2015) [64] did apply the method in [63] in studying gender diversity on GitHub. However, instead of performing an independent validation, they referred the precision results from [63].

Given that the study is only targeting for the top female developers, we decide to identify all female developers in the top 5,000 list through human judgment. First, a researcher manually examined all profiles of the top 5,000 developers on GitHub to determine their gender identifications. In this process, she utilizes GitHub users' profiles (e.g., name, picture, etc.), linked website, and linked social media profile (e.g., twitter, Facebook, Linkedin, Weibo, Zhihu, etc.) to form her judgments. The other two researchers cross-validate the list of identified female developers and exclude 8 falsely identified ones. We reach an agreement on a list of 168 females developers. After identifying all female developers, we randomly sample the same number ($N = 168$) male developers for the comparison with the female developers in the data analysis. In total, we have a sample contains 336 individual subjects.

2012, 17, 1, 0
2012, 18, 3, 0
2012, 19, 3, 0
2012, 20, 4, 0
2012, 21, 3, 2
2012, 22, 0, 0

Figure 3: An excerpt of a subject's data.

4.3 Data Preparation

We use the SQL queries based on subjects' login to extract data from GHTorrent. First, we use a joint SQL query to extract the followers data for each subject from the GHTorrent, which consists all his or her followers' usernames and the timestamps of the following relationship established. Second, since we only interest in the subject's first attempt to contribute to a project, we query each subject' all pull requests to **new** projects. For each subject, we store the two query results into separated `.csv` files. Finally, through a `Python` script, we grouped the each subject's two data file according to the calendar week (starting from every Monday) which implicitly reflects her weekly working cycle.

Each sampled subjects' quantitative data is an ordered list of 4-tuple in the following form:

`<year, week, new followers, pull request to new projects>`

Fig. 3 shows an excerpt of a subject's data. Let's take `<2012, 21, 3, 2>` as an example (marked with gray color in fig. 3. "2012" is the year. "21" means the 21st week of 2012. "3" indicates the number of the subject's new followers in this week while "2" shows the number of her pull requests to new projects in the same week. If there is no new followers and new pull request in a week, we simply fill the third and the fourth elements with "0." Thus, it is easy to generate two equal-length time series: one for the number of new followers by week, and the other for the number of pull request to new projects by week. Fig. 4 presents examples of these two time series. For the reliability of the quantitative analyses, we remove the subjects who initiate pull requests to less than 5 different repositories. This results in a sample consisting of 145 valid female data cases and 153 valid male data cases (total cases: 298).

5 DATA ANALYSIS METHOD

To establish the relationship between the follower growth and the pull request to new projects, we conduct a set of quantitative analysis. First, for each identify top developer (both male and female developers), we performed the Granger causality test on the time series of new pull requests and the time series of acquiring followers. Then, we conduct the Chi-square test of Goodness-of-Fit to examine the difference between the male and female sub-samples. All statistical analyses are performed with `R 3.4.1` [51] for macOS High Sierra (version 10.13.1). We follow the American Statistical Association's principles to interpret the statistical results [65].

5.1 Granger Causality Test

In his seminal paper, Nobel Prize in Economic Science Laureate Clive Granger [24] developed the idea of Granger causality. The notion of Granger causality is simple: for two time series X and Y, if lagged values of X help predict current values of Y in a forecast formed from lagged values of both X and Y, then X is said to Granger cause Y. In this study, we want to investigate if initiating a pull request to a new repository is "Granger" caused by the fast increase of followers in a prior period. Hence, the two time series X and Y for analysis are each subject's dynamics of followers and dynamics of pull requests to new repositories. We have developed constructed these two time series in the data preparation step (see section 4.2 and Fig. 4). Without the loss of generality, let's name the two series as: *Follower* and *PR*. To perform Granger causality test, the following model is estimated with the Ordinary Least Squared (OLS) method:

$$PR_t = \mu + \sum_{i=1}^{L} \alpha_i PR_{t-i} + \sum_{i=1}^{L} \beta_i Follower_{t-i} \quad (1)$$
$$L = max.\ no.\ of\ lags,\ L = 4\ in\ our\ study$$

The model specified in Eq. 1 contains two parts. The first is the auto-regressive part ($\sum_{i=1}^{L} \alpha_i PR_{t-i}$), which indicates the relationship with its own prior periods. The second is the causal part ($\sum_{i=1}^{L} \beta_i Follower_{t-i}$) related to the other time series, i.e., dynamics of followers. Obviously, the null hypothesis (H_0) is: $\beta_1 = \ldots = \beta_L = 0$ which means that the pull request at time t has nothing with the changes of followers in prior periods. Fig. 4 show an example of two time series where the Granger causality test is significant.

One important issue in Granger causality test is determining the optimal "L", and we design and conduct a small experiment to determine the optimal lag. We randomly select 50 data cases. For each, we run Granger causality test with lags in the range of [1, 8]. 17 cases (highest in all 8 possible lags) achieve the best AIC when $L = 4$ while 42 cases achieve best AIC when the maximal lag is in the range of [1, 4].

6 RESULTS AND FINDINGS–SOCIAL ATTRACTION MATTERS!

We report the results and findings from the quantitative analyses in this section. They provided general support to the two predictions derived from the theoretical proposition. The predictions include the existence of the competence-confidence gap. Thus, the validity of the theoretical proposition is supported in the context of GitHub.

For each subject in our sample, we run an independent Granger causality test with the maximum lag (L in Eq. 1) is four, which means we consider the impact of follower number's change within four

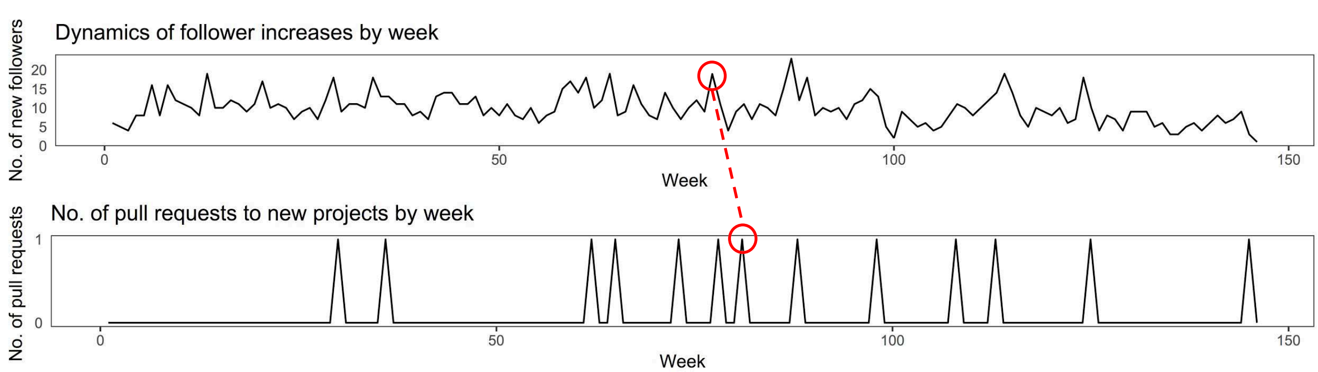

Figure 4: A subject's (Fig.5-subject 23 in the rank 1-1000) two time series showing Granger Causality. Please note that, we only show the data from 2015 to 2017 to make the plot more readable. Note that the peaks in the pull request (the second plot) often appear in a few weeks after the local peaks in the dynamics of followers (the first plot).

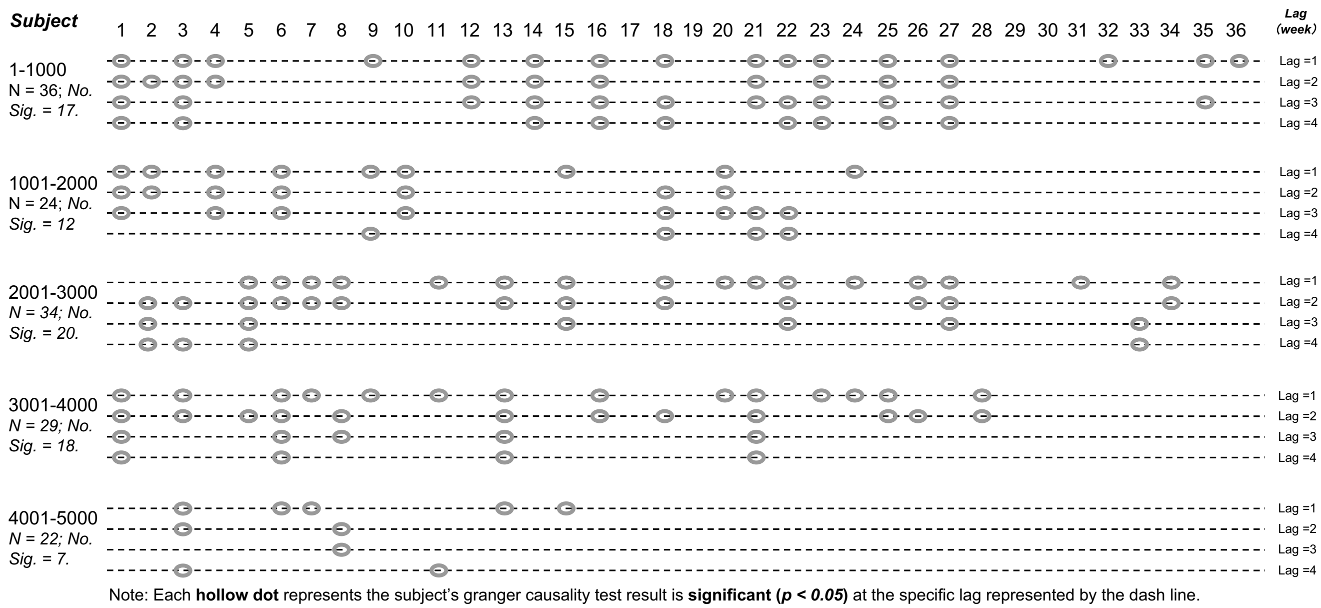

Figure 5: The results of Granger causality tests for the 145 top female developers with lag up to 4 weeks). The hollow dots denote significant at the 0.05 level. For the convenience of visualizing the results, we organize them into 5 subgroups according to their rankings in the top 5,000 list.

weeks. In total, we perform 298 (145: female, 153: male) independent Granger causality tests.

Fig. 5 reports the Granger causality test results for the 145 female developers. In this diagram, we organize the sample into five subgroups based on intervals of one thousand follower ranks. In each subgroup, there are four horizontal dash lines representing 4-week intervals when the lag in the range of [1, 4], and each vertical set represents an individual subject. If there is a hollow dot on the dash line of a specific lag, it indicates that the Granger test result of this subject is significant at the lag order of i (in Eq. 1 $\beta_i \neq 0$). The "significance" means for this subject, followers increase in the week $t-i$ are likely to Granger cause pull requests to new repositories in the week t. Let's take the 8th subjects in the third subgroup as an example, there are two hollow dots lying on the two dash line for lag 1 and lag 2. It indicates that the subject's pull request in week t is Granger caused by the follower increase happened in the week $t-1$ and $t-2$.

In total, 73 female data cases are significant at least on one of 4 possible lags in the Granger causality test (at least one $p < 0.05$ for $\beta_i \neq 0$). In the first (1-1000) and second (1001-2000) subgroups, about half of the subjects are significant (17 out of 36: 47.2% and 12 out of 24: 50%). The third and forth group of subjects have a slightly higher significant rate (20 out of 34: 58.8% and 18 out of 29: 62.1%). However, the last subgroup has the lowest significant rate (7 out of 22: 31.8%). This is because there are too many "0" in the time series of follower increase by week which makes the Granger test maybe not effective. For all these 73 cases that have at least one significant, there are 160 significant βs. Almost all of them are greater than 0 (154 out of 160: 96.25%). It suggests that the fast increase of followers has a positive effect on making a pull request to a new project. Moreover, it is obvious that week $t-1$ and $t-2$ have a stronger effect than weeks $t-3$ and $t-4$ for they account for the majority of significant results (102 out of 160: 63.75%).

For comparing the differences between the male and female sub-samples, we perform the Granger causality test on 153 male data cases. There are only 26 out of 153 (17%) male data cases have significant results, much less than the female ones (73 out of 145: 50.3%). We do not visualize the results as what we do for the female data cases (Fig. 5) to keep the conciseness of the paper. To statistically examine the differences, we code each individual's Granger causality test results into a 0-1 variable (0: Not Sig., 1: Sig.) and put it into corresponding gender groups. For female developers, there are 73 "1" and 72 "0". For male developers, there are 26 "1" and 127 "0". It is obvious that both are Bernoulli sequences. Then, we use Chi-square test of Goodness-of-Fit to test whether the two sub-samples follow the same Bernoulli distribution. The results strongly support that the results of male and female sub-samples are from different Bernoulli distributions ($\chi^2 = 119.26$, $df = 1$, $p-value = 2.2e-16$).

7 DISCUSSION

7.1 Inherited Bias towards Female Developers

Based on our dataset of top 5000 developers, we found that only a very small percentage (~3%) of developers are female. The whole culture and social institution are male-dominated [38]. Although the strategies summarized in section 6.2 are proven effective, an unpleasant fact is that the most of these proven effective strategies adopted by the top female developers are catering to the existing, male-dominated social norms and historical prejudice towards women. By respecting gender social norms and fitting themselves into the gender stereotypes, they are rewarded in some way, e.g., have more followers. In fact, this type of follower/following relationship is similar to that on Twitter or Instagram. Doing so actually reinforces the bias and prejudice towards female developers. There is a natural tendency for people to behave according to their perceived social norm rather than their own belief [58]. To alter this situation requires tremendous effort to reshape the existing social value and norms [46]. There is still a long way to go for the software development industry to appreciate women's competence and technical contributions.

7.2 Strategies to Overcome the Gap

Top female developers use multiple strategies to overcome the competence-confidence gap. Compared with the men, women tend to care more about altruistic reputation accumulating by helping others and are often judged by such behaviors [17, 67]. Our observation suggests that top female developers take advantage of that to improve their reputation. One of the major actions is participating the knowledge-sharing projects which produce documentation, tutorials, open textbooks, and others to help the community but do not produce software systems [36]. By participating these projects, they gain the reputation as helpers in the community, which fits the gender social expectation. Moreover, these knowledge-sharing projects are less technical, so even when the female developers underestimate their capability, they may still feel confident to contribute to these projects. Most of such projects do not adopt pull request model. Thus they are less competitive without extra barriers to contribute. Women, who often hold themselves back in competitive environment [45], would feel more comfortable to contribute to these projects.

These projects, although are often being neglected, are important for the software development [15], and appreciated by the many developers, particularly the newcomers [56]. Hence, female developers can receive a lot of attention in a relatively low barrier way through launching or participating knowledge-sharing projects. Their followers may increase and hence social attractions may be improved. Thus, a female developer will have more confidence to contribute to a new competitive project.

A more comprehensive qualitative study on publicly available behavioral traces and user interviews is expected in future work. We plan to explore multiple strategies used by top female developers to overcome the confidence gap, and the responses from the community. The follow-up study would help self-underestimated female developers identify immediately applicable ways to participate the community more proactively.

7.3 Design Implications

The current design of GitHub does not take gender differences into account. The reputation mechanisms and user interfaces are identical for both male and female users. As we mentioned before, the literature suggests that women are more concerning about altruistic reputations [67]. The altruistic reputations are often neglected and less recognized by the community. For example, a user spending a lot of efforts to write beginner's guide for `Reactive.js` perhaps does not gain the comparable level of reputation compared with those who spend equal time on contributing code to `Reactive.js`. It may be worth to try implementing a reputation system based on altruistic contributions. Pull-request model has well facilitated scaling up, but it gives advantages to men who are tend to prefer a more competitive environment and creates barriers women who care more about helping each other in a less competitive environment. How to improve the pull request model to ensure the gender fairness is an open challenge. An immediate but not complete remedy may be designing a reputation system based on pro-social behaviors and highlighting it in a user's profile.

Another design alternative is to leverage behavioral economics theories to build adaptive user interface for women. Prospect theory [35] claims that people are less sensitive to their gains than their losses. Since female developers' confidence to contribute to a new project related to the increase of followers. We can design interfaces that highlight the "gain" of new followers, for example, adding a widget repeatedly showing the names of new followers added in last week to a female user's own profile page. Through such an intervention, her "gains" may be reinforced, hence it may help to boost her confidence towards contributing to a new repository. Moreover, our results suggest that only follower increases in the past two weeks have significant effects in most cases, partially because our cognition and memory system cannot trace back too much. It might be helpful to design some user experience to help the users recall the follower increase happens more than two weeks ago. Using behavioral economics in user interaction design is an emerging paradigm in human-computer interaction community [29, 41, 68]. We believe that it can help to tackle the competence-confidence gap.

7.4 Methodological Implications

One of the contributions of the paper lies in its methodological value. Software engineering is increasingly concerned with "theory" because the theoretical knowledge provides a crucial counterpoint to the practical knowledge represented by specific processes and techniques [52]. Among many types of theories, theories of scientific explanation–the elucidation of causal relationships among variables or events–are the central aim [48]. Particularly, for software development processes consist of events/activities which have sequential/temporal features. Reasoning causality among these events has its unique value on developing solid theoretical software engineering knowledge. However, establishing causality is often very hard [49].

The method we used in this paper, propose an appropriate way to extract causal relationship from empirical data that has temporal features[2]. The method has two steps. In the first step, researchers may start from existing theories, e.g., confidence-competence gap and social attraction model in this paper, then deduce theoretical expectations or hypotheses in a specific scope, e.g. P1 and P2 in section 3.1. The second step contains collecting relevant data within the scope and using proper statistical techniques to validate

[2]We do not cover the epistemological discussion in this paper. Instead, we take a pragmatic position to focus on introducing the method. Interested readers may refer Carl Hempel's classics book [30]

the propositions. If both events (cause and effect) are time-related and can be represented as panel data, Granger causality is often a good choice. The method has potential to be used to reasoning causality for a wide range of phenomena in software engineering practices. The theory development and statistical analyses give a live example of how to apply it. For instance, let's suppose a researcher attempts to build causality between positive emotion in software development teams and productivity improvements. She may first pick some existing theories related to emotion and team performance e.g., [2]. Then, she should carefully apply them in the context software development teams to deduce theoretical predictions/hypotheses. Then, for each team in her dataset, she can construct two time series, one is team's emotion dynamics and the other is periodically measured productivity indicators. By applying Granger causality test, she may either accept or reject the hypotheses. The acceptance of the hypotheses indicates the establishment of the causality while the rejection means the causality may not valid. However, as every research methods for causality inference, the method and corresponding statistical analyses used in our paper have its own limitations and restrictions [25]. Hence, We urge caution for applying the method and Granger causality test to develop causal relationships in readers' own research.

7.5 Threats to Validity

Any empirical study is subject to a number of threats to validity. Validity of extraction of information from GitHub data can be threatened in a way similar to digital traces [33] or put in "peril" by concerns related to mining GitHub data [36]. We expended a great deal of effort to avoid the threats above by various means. For example, we use human judgment on multiple data source include linked external information rather than apply machine learning methods to identify the top female developers on GitHub, and perform cross-examination by different researchers. Focusing on pull requests helps us to avoid multiple perils in [36]. Furthermore, to reduce the threats related to analysis methods[7], we use well-established time-series analysis and other statistical methods. The interpretations of the statistical analyses follow ASA's guideline and principles. We provide a discussion on the relationship between "true causality" and "Granger causality" in the study design and gives further discussions on methodological issues related to causality inferences in section 7.4.

We use manual effort to identify the top female developers. The decision is made based on several practical considerations. First, even the automated methods are usually high in precision, their performance on recall is not quite well. Given the fact that there is only a very small proportion of the female developers in the top-5,000 list, they cannot help us develop a dataset that is meaningful for any statistical analysis. Second, a substantial proportion of the top female developers are from East Asia, particularly China[3]. Their social media profiles are often in their native language, hence, gender inference methods based on social profiles [60] may not work well. Last, a number of developers gender identity requires substantial human judgments on very subtle details. For example, a male developer's profile picture, blog, and weibo (a China microblogging service) all suggest that he is a "woman". The only cue enabling us to correctly identify his gender is a sentence disclosing that he is a crossdresser [伪娘]. For our study only focuses on a few hundreds of top female developers, the scale of the data is not our major concern. Hence, manually identifying the top female developers is the best available choice and guarantee the validity. The control group of male developers is constructed through one-shot random sampling. For the purpose of comparisons, it is sufficient because we use the whole sample of female developers in the top-5,000 list without any selective filtering. However, there are some fine-grained techniques to improve the process of sampling the control groups. These techniques include controlling factors such as tenure in GitHub, location, etc. Multiple random sampled control groups for repeated analysis will also help to improve the already high validity.

[3]China continuously has high female workforce participation rate even compared with the industrial economies, see https://data.worldbank.org/indicator/SL.TLF.CACT.FE.ZS.

8 RELATED WORK

Beyond the popular media and statistics from public surveys, formal research also confirms the situation that women are underrepresented in the software development industry, and in almost all major economies. Ironically, information technology, which was viewed as an opportunity for women in early 2000s, reproduces and strengthen gender inequalities seen in the broader fabric of society [53]. For example, female staffs in large software companies in Germany were assigned to support positions, such as project management or quality assurance, while their STEM backgrounds qualify them for higher-paid technical positions. To help remedy this situation, researchers have done a great deal of work to promote women's participation in software development. For example, Gorriz and Medina [22] suggested that software games can promote women's engagement in computing. Gweon et al. [28] explores design concepts and principles to engage girls in learning programming through different media and interaction techniques. Studies such as [11] attempt to identify the social and psychological factors deterring women's participation. Retention of female developers is also often discussed as an issue [12].

Software engineering researchers have developed engineering approaches to help improve the inclusiveness of women in computing. Burnett and her colleagues have developed a family of Gender HCI techniques to make the end-user development more gender inclusive, e.g., [4]. Recently, they proposed GenderMag approach and applied it to software product teams in Microsoft [9]. It helps software professionals realize and evaluate gender inclusiveness of the software that they are building. A longitudinal study on the effectiveness of this method is reported in [8]. Inspired by research in Machine Learning area, fairness test, a newly presented idea of testing software for discrimination, could also help with challenges of inclusiveness although it is designed for multiple type of biases[20].

Besides, there are empirical evidence clearly demonstrate that increasing gender diversity in software teams does not only have its ethnic/moral necessity but also lead to pragmatic values. Vasilescu et al. [64] shows that gender diversity has positive influence on team's productivity. Aué et al. [1] suggests similar results that social diversity does contribute to project growth. While these studies prove the benefits of increasing gender diversity, a reluctant fact is that females' participation is still quite peripheral [18]. Moreover, a recent study confirms that over 80% of the barrier types for newcomer include attributes that are biased against women [31]. Hence, creating a path and lowering barriers for female developers' self-guided personal development is critical for them to achieve their own technical distinctions and for the community to benefit from gender diversity.

9 CONCLUSIONS

Encouraging participation of female software developers in projects is an important goal. Understanding the causes of the low participation will help us to find better solutions. In this paper, we develop a theoretical explanation for why well-qualified female developers have a low rate of initiating pull requests to a new project under GitHub's pull request model. The theoretical explanation features the competence-confidence gap among female developers and the moderating role of social attraction. We designed and conducted a mixed-method empirical study to validate the theoretical explanation. Further, our analyses also identify the strategies used by the top female developers on GitHub to overcome the competence-confidence gap by achieving greater social attraction within the community. Our results also indicate a set of mechanisms and user experience design implications that can support female developers in overcoming the gap. To the best of our knowledge, this is the first work that provides a theoretical explanation and empirical evidence for female developers' low rate of attempting to contribute to new projects on GitHub through the lens of the competence-confidence gap and social attraction. The study opens rich future research opportunities of leveraging mechanism and user experience design to help female developers overcome the competence-confidence gap. We plan to continue our work in these directions. Moreover, our work immediately leads to a critical question about the current software engineering education practices, which are gender insensitive and only focus on helping students develop technical skills. That is: how to design innovative educational programs that help female students build confidence? Researchers, including ourselves, should continue to push the research in this direction.

We acknowledge that the problem investigated in this study is only a small part of the broader problem of empowering women who are historically social and economically undermined in male-dominated society. Fixing the problem requires tremendous efforts from the whole world and may take a few generations. We would like to end the paper with Annie-Marie Slaughter's message in [55]: "*If women are ever to achieve real equality as leaders, then we have to stop accepting male behavior and male choices as the default and the ideal. We must insist on changing social policies and bending career tracks to accommodate our choices, too.*"

ACKNOWLEDGMENTS

This work is partially supported by NSF CNS-1757680 and a Ford Foundation grant for Inclusive Economics. We thank the reviewers and others who provided critical feedback on this work, including our colleague, Dr. Bonnie Ruberg. Of course any remaining deficiencies are our own.

REFERENCES

[1] Joop Aué, Michiel Haisma, Kristín Fjóla Tómasdóttir, and Alberto Bacchelli. 2016. Social Diversity and Growth Levels of Open Source Software Projects on GitHub. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '16)*. ACM, New York, NY, USA, Article 41, 6 pages. https://doi.org/10.1145/2961111.2962633
[2] Paul G Bain, Leon Mann, and Andrew Pirola-Merlo. 2001. The innovation imperative: The relationships between team climate, innovation, and performance in research and development teams. *Small group research* 32, 1 (2001), 55–73.
[3] Manuela Ed Barreto, Michelle K Ryan, and Michael T Schmitt. 2009. *The glass ceiling in the 21st century: Understanding barriers to gender equality.* American Psychological Association.
[4] Laura Beckwith and Margaret Burnett. 2004. Gender: An important factor in end-user programming environments?. In *Visual Languages and Human Centric Computing, 2004 IEEE Symposium on*. IEEE, 107–114.
[5] Carlo Berzuini, Philip Dawid, and Luisa Bernardinell. 2012. *Causality: Statistical Perspectives and Applications.* John Wiley & Sons.
[6] Wiebe E Bijker, Thomas P Hughes, Trevor Pinch, and Deborah G Douglas. 2012. *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology.* MIT press.
[7] Christian Bird, Peter C. Rigby, Earl T. Barr, David J. Hamilton, Daniel M. German, and Prem Devanbu. 2009. The Promises and Perils of Mining Git. In *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories (MSR '09)*. IEEE Computer Society, Washington, DC, USA, 1–10.
[8] Margaret Burnett, Robin Counts, Ronette Lawrence, and Hannah Hanson. 2017. Gender HCI and Microsoft: Highlights from a Longitudinal Study. In *Proceedings of the 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2017)*. 139–143.
[9] Margaret Burnett, Simone Stumpf, Jamie Macbeth, Stephann Makri, Laura Beckwith, Irwin Kwan, Anicia Peters, and William Jernigan. 2016. GenderMag: A method for evaluating software's gender inclusiveness. *Interacting with Computers* 28, 6 (2016), 760–787.
[10] Tomas Chamorro-Premuzic. 2013. Why Do So Many Incompetent Men Become Leaders? Available at https://hbr.org/2013/08/why-do-so-many-incompetent-men. (2013). [Online; accessed 09/26/2017].
[11] Sapna Cheryan, Victoria C Plaut, Paul G Davies, and Claude M Steele. 2009. Ambient belonging: how stereotypical cues impact gender participation in computer science. *Journal of personality and social psychology* 97, 6 (2009), 1045.
[12] J. McGrath Cohoon. 2001. Toward Improving Female Retention in the Computer Science Major. *Commun. ACM* 44, 5 (May 2001), 108–114. https://doi.org/10.1145/374308.374367
[13] James Damore. 2017. Google's Ideological Echo Chamber-How bias clouds our thinking about diversity and inclusion. Available at https://goo.gl/sH4soj. (2017). [Online; accessed 09/26/2017].
[14] datausa.io. 2015. Demographics of Software Developers (2015). https://datausa.io/profile/soc/15113X/. (2015). [Online; accessed 09/26/2017].
[15] Sergio Cozzetti B. de Souza, Nicolas Anquetil, and Káthia M. de Oliveira. 2005. A Study of the Documentation Essential to Software Maintenance. In *Proceedings of the 23rd Annual International Conference on Design of Communication: Documenting &Amp; Designing for Pervasive Information (SIGDOC '05)*. ACM, New York, NY, USA, 68–75.
[16] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. 2008. Selecting empirical methods for software engineering research. *Guide to advanced empirical software engineering* (2008), 285–311.
[17] Nancy Eisenberg, Randy Lennon, and Karlsson Roth. 1983. Prosocial development: A longitudinal study. *Developmental Psychology* 19, 6 (1983), 846.
[18] Casey Fiesler, Shannon Morrison, R. Benjamin Shapiro, and Amy S. Bruckman. 2017. Growing Their Own: Legitimate Peripheral Participation for Computational Learning in an Online Fandom Community. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 1375–1386. https://doi.org/10.1145/2998181.2998210
[19] Alyssa Frazee. 2014. Exploring the data on gender and GitHub repo ownership. Available at http://alyssafrazee.com/2014/06/04/gender-and-github-code.html. (2014). [Online; accessed 09/26/2017].
[20] Sainyam Galhotra, Yuriy Brun, and Alexandra Meliou. 2017. Fairness Testing: Testing Software for Discrimination. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017)*. ACM, New York, NY, USA, 498–510. https://doi.org/10.1145/3106237.3106277
[21] Yaping Gong, Jia-Chi Huang, and Jiing-Lih Farh. 2009. Employee learning orientation, transformational leadership, and employee creativity: The mediating role of employee creative self-efficacy. *Academy of management Journal* 52, 4 (2009), 765–778.
[22] Cecilia M. Gorriz and Claudia Medina. 2000. Engaging Girls with Computers Through Software Games. *Commun. ACM* 43, 1 (Jan. 2000), 42–49.
[23] Georgios Gousios. 2013. The GHTorent Dataset and Tool Suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories (MSR '13)*. IEEE Press, Piscataway, NJ, USA, 233–236.

[24] Clive WJ Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society* 37, 3 (1969), 424–438.

[25] Clive WJ Granger. 1980. Testing for causality: a personal viewpoint. *Journal of Economic Dynamics and Control* 2 (1980), 329–352.

[26] Clive W.J. Granger. 2004. Time Series Analysis, Cointegration, and Applications. *American Economic Review* 94, 3 (June 2004), 421–425.

[27] Laura Guillén, Margarita Mayo, and Natalia Karelaia. 2016. The competence-confidence gender gap: Being competent is not (always) enough for women to appear confident. *Journal of Applied Psychology, In Submission* (2016).

[28] Gahgene Gweon, Jane Ngai, and Jenica Rangos. 2005. Exposing Middle School Girls to Programming via Creative Tools. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'05)*. Springer-Verlag, Berlin, Heidelberg, 431–442.

[29] John Harvey, David Golightly, and Andrew Smith. 2014. HCI As a Means to Prosociality in the Economy. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2955–2964.

[30] Carl Hempel. 2009. Scientific Explanation. *Philosophy of Science: An Historical Anthology* (2009), 344.

[31] Claudia Hilderbrand, Charles Hill, Anita Sarma, Amber Horvath, Logan Simpson, Nupoor Patil, Zoe Steine-Hanson, Hema Susmita Pedala, Christopher Mendez, and Margaret Burnett. 2017. Open Source barriers to entry, revisited: A tools perspective. *Technical report (Oregon State University. Dept. of Computer Science)* (2017).

[32] Michael A Hogg and Elizabeth A Hardie. 1991. Social Attraction, Personal Attraction, and Self-Categorization-, A Field Study. *Personality and Social Psychology Bulletin* 17, 2 (1991), 175–180.

[33] James Howison, Andrea Wiggins, and Kevin Crowston. 2011. Validity issues in the use of social network analysis with digital trace data. *Journal of the Association for Information Systems* 12, 12 (2011), 767.

[34] Nicole Immorlica, Greg Stoddard, and Vasilis Syrgkanis. 2015. Social Status and Badge Design. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 473–483.

[35] Daniel Kahneman and Amos Tversky. 1979. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the econometric society* (1979), 263–291.

[36] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. 2014. The Promises and Perils of Mining GitHub. In *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014)*. ACM, New York, NY, USA, 92–101.

[37] Katty Kay and Claire Shipman. 2014. The Confidence Gap. https://www.theatlantic.com/magazine/archive/2014/05/the-confidence-gap/359815/. (2014). [Online; accessed 20-October-2017].

[38] Mary Kirk. 2009. Language as Social Institution: The Male-Centered IT Culture. In *Gender and Information Technology: Moving Beyond Access to Co-Create Global Partnership*. IGI Global, 119–142.

[39] Barbara A. Kitchenham, Shari Lawrence Pfleeger, Lesley M. Pickard, Peter W. Jones, David C. Hoaglin, Khaled El Emam, and Jarrett Rosenberg. 2002. Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Trans. Softw. Eng.* 28, 8 (Aug. 2002), 721–734.

[40] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. 2008. A Few Chirps About Twitter. In *Proceedings of the First Workshop on Online Social Networks (WOSN '08)*. ACM, New York, NY, USA, 19–24.

[41] Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. 2011. Mining Behavioral Economics to Design Persuasive Technology for Healthy Choices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 325–334.

[42] Elinor M. Madigan, Marianne Goodfellow, and Jeffrey A. Stone. 2007. Gender, Perceptions, and Reality: Technological Literacy Among First-year Students. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07)*. ACM, New York, NY, USA, 410–414.

[43] Meredith Meyer, Andrei Cimpian, and Sarah-Jane Leslie. 2015. Women are underrepresented in fields where success is believed to require brilliance. *Frontiers in Psychology* 6, Article 235 (2015).

[44] Paul Miller. 2013. Most active GitHub users. Available at https://gist.github.com/paulmillr/2657075. (2013). [Online; accessed 09/26/2017].

[45] Muriel Niederle and Lise Vesterlund. 2007. Do women shy away from competition? Do men compete too much? *The Quarterly Journal of Economics* 122, 3 (2007), 1067–1101.

[46] Karine Nyborg, John M Anderies, Astrid Dannenberg, Therese Lindahl, Caroline Schill, Maja Schlüter, W Neil Adger, Kenneth J Arrow, Scott Barrett, Stephen Carpenter, et al. 2016. Social norms as solutions. *Science* 354, 6308 (2016), 42–43.

[47] Greg R Oldham and Anne Cummings. 1996. Employee creativity: Personal and contextual factors at work. *Academy of management journal* 39, 3 (1996), 607–634.

[48] Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge university press.

[49] Dewayne E Perry. 2015. A framework for exploring unifying theories of empirical software engineering. *Science of Computer Programming* 101 (2015), 99–123.

[50] Eva M Pomerantz, Ellen Rydell Altermatt, and Jill L Saxon. 2002. Making the grade but feeling distressed: Gender differences in academic performance and internal distress. *Journal of Educational Psychology* 94, 2 (2002), 396.

[51] R Development Core Team. 2008. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org ISBN 3-900051-07-0.

[52] P. Ralph. 2018. Toward Methodological Guidelines for Process Theories and Taxonomies in Software Engineering. *IEEE Transactions on Software Engineering* (2018), To Appear. https://doi.org/10.1109/TSE.2018.2796554

[53] Esther Ruiz Ben. 2007. Defining expertise in software development while doing gender. *Gender, Work & Organization* 14, 4 (2007), 312–332.

[54] Brian Skyrms and William L Harper. 2012. *Causation, Chance and Credence: Proceedings of the Irvine Conference on Probability and Causation*. Vol. 1. Springer Science & Business Media.

[55] Anne-Marie Slaughter. 2015. *Why Women Still Can't Have It All*. OneWorld.

[56] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David F. Redmiles. 2015. Social Barriers Faced by Newcomers Placing Their First Contribution in Open Source Software Projects. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW 2015, Vancouver, BC, Canada, March 14 - 18, 2015.* 1379–1392.

[57] Loulou Storey. 2017. The Confidence Gap: A Guide To It And How To Reduce It. Available at http://www.huffingtonpost.co.uk/loulou-storey/the-confidence-gap-a-guid_b_18171592.html. (2017). [Online; accessed 09/26/2017].

[58] Margaret E Tankard and Elizabeth Levy Paluck. 2016. Norm perception as a vehicle for social change. *Social Issues and Policy Review* 10, 1 (2016), 181–211.

[59] Cheryl Tay, Soon Ang, and Linn Van Dyne. 2006. Personality, biographical characteristics, and job interview success: a longitudinal study of the mediating effects of interviewing self-efficacy and the moderating effects of internal locus of causality. *Journal of Applied Psychology* 91, 2 (2006), 446.

[60] Josh Terrell, Andrew Kofink, Justin Middleton, Clarissa Rainear, Emerson Murphy-Hill, Chris Parnin, and Jon Stallings. 2017. Gender differences and bias in open source: Pull request acceptance of women versus men. *PeerJ Computer Science* 3 (2017), e111.

[61] Jason Tsay, Laura Dabbish, and James Herbsleb. 2014. Influence of Social and Technical Factors for Evaluating Contribution in GitHub. In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. ACM, New York, NY, USA, 356–366.

[62] Jason Tsay, Laura Dabbish, and James Herbsleb. 2014. Let's Talk About It: Evaluating Contributions Through Discussion in GitHub. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014)*. ACM, New York, NY, USA, 144–154.

[63] Bogdan Vasilescu, A. Capiluppi, and A. Serebrenik. 2012. Gender, Representation and Online Participation: A Quantitative Study of StackOverflow. In *2012 International Conference on Social Informatics (SocialInfo. '12)*. OUP, 332–338.

[64] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark G.J. van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. 2015. Gender and Tenure Diversity in GitHub Teams. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3789–3798.

[65] Ronald L. Wasserstein and Nicole A. Lazar. 2016. The ASA's Statement on p-Values: Context, Process, and Purpose. *The American Statistician* 70, 2 (2016), 129–133. https://doi.org/10.1080/00031305.2016.1154108 arXiv:https://doi.org/10.1080/00031305.2016.1154108

[66] James Wolf. 2011. A Reexamination of Gender-based Attitudes Toward Group Projects: Evidence from the Google Online Marketing Challenge. *Comput. Hum. Behav.* 27, 2 (March 2011), 784–792. https://doi.org/10.1016/j.chb.2010.11.001

[67] Lynne Zarbatany, Donald P Hartmann, Donna M Gelfand, and Paul Vinciguerra. 1985. Gender differences in altruistic reputation: Are they artifactual? *Developmental Psychology* 21, 1 (1985), 97.

[68] Yunfeng Zhang, Rachel K.E. Bellamy, and Wendy A. Kellogg. 2015. Designing Information for Remediating Cognitive Biases in Decision-Making. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2211–2220.