Effectiveness of Conflict Management Strategies in Peer Review Process of Online Collaboration Projects

Anonymized for review

ABSTRACT

In online collaboration projects, conflicts often arise in the peer reviewer process, due to the disagreement on whether one's contribution can be accepted. These conflicts generally have detrimental effects on the contributors' continuous participation in the communities. However, few studies investigated how to manage these conflicts appropriately. This paper aims to examine the effectiveness of three strategies – rational explanation, constructive suggestion, and social encouragement – in managing conflicts. In the analysis of 170 online software development projects, we investigated how the different conflict management strategies that aimed at handling contributors' arguments in the peer review process influenced their continuous participation in the projects. The results show that (i) conflicts significantly increase contributors' likelihood of leaving the communities. (ii) Neither providing rational explanations nor socially encouraging could reduce the negative consequences of conflicts. (iii) Only providing constructive suggestions has a positive effect in retaining the contributors.

Author Keywords

online collaboration, conflict management, survival analysis.

INTRODUCTION

Online open collaboration projects rely on a large group of volunteer contributors to collectively produce important artifacts or services. Peer review is an important mechanism to ensure the quality of the products [33]. Because of the large variance in background, viewpoints and experience among contributors, conflicts often arise in the peer review process due to the disagreement on whether one's contribution should be accepted [42]. For example, Joode et al. found that both task conflict and affective conflict can occur in open source communities, especially when one's software patch was unjustly rejected [41].

Conflicts generally have negative effects on group loyalty, work productivity, and job satisfaction [10, 19, 21]. Although the prior CSCW research has identified the potential detrimental effect of conflicts on the continuous success of online collaboration projects [15, 24, 25], very few of them studied how to manage the conflicts and disagreement appropriately.

Research in the traditional organization settings proposed variant conflict management strategies that could serve the purpose of minimizing the negative consequences of conflict [4, 5]. For example, Blake and Mouton proposed five methods to handle conflicts: forcing, withdrawing, smoothing, compromising, and problem solving [7]. Rahim and Bonoma also summarized five styles of handling conflict (i.e., integrating, obliging, dominating, avoiding and compromising) and the situations in which these are appropriate [31]. However, it remains unknown whether these strategies are still effective in online context.

In this paper, we want to ask the following research question: how to handle online contributors' conflicts in the peer review process in order to minimize the negative consequences on contributors' continuous participation? Combining existing conflict management theories in traditional organizations and our domain knowledge of online collaboration projects, we identified three conflict-management strategies: rational explanation, constructive suggestion and social encouragement. We tested the effectiveness of these three strategies in managing the conflicts which arise in GitHub projects' peer review process. Additionally we tested whether the status of the person who delivered the conflict-management strategies affects the effectiveness of the strategies.

The result shows that when conflict happened, contributors tend to leave the project. Neither rational explanations nor social encouragement had an effect on retaining contributors in the project. Only providing constructive suggestions reduced the negative consequence of conflict and resulted in higher likelihood of staying in the project. Additionally, we found that whether the people who delivered the conflict management strategies were the project's administrators or not had no influence on the contributor's continuous participation. We also conduct an

open-end user survey to help us understand the quantitative findings. Based on our findings, we suggest designing tools or mechanisms to promote the awareness of conflict in online collaboration projects and to more effectively manage the conflicts.

RELATED WORK AND HYPOTHESES

We ground our study in researches on conflict and its management theories in both traditional organizations and online collaboration settings. In the following sections, we will review related work and propose our hypotheses accordingly.

Conflicts in Offline Settings

Conflicts are common in people's everyday life. Rahim described conflict as "*an interactive process manifested in incompatibility, disagreement, or dissonance within or between social entities (i.e., individual, group, organization)*" [4]. Early researches on conflict concentrated on situations in which members have opposing goals and assumed a basic conflict of goals within the group [10, 39]. But, later studies suggested that people may have conflicts even when they generally agree on goals and conflict may develop from people's attempts to cooperate or coordinate their efforts [22].

Conflicts in Peer Review of Online Collaboration Project

Conflicts occur frequently in online collaboration settings due to a lack of shared context, limited information sharing channels, and weak interpersonal bonds between members [20]. Conflicts are especially common in the peer review process, in which people decide whether one's contributions should be accepted or not [42]. There are different forms of peer review in different online collaboration settings. In many software development projects, there is a formal peer review system where contributors propose code changes (e.g., pull-requests in GitHub) and reviewers decide whether the contribution should be accepted or not [40]. In Wikipedia, both formal peer review and informal peer review exists. For instance, featured articles (FA) are determined by a formal peer review process where group of editors decide whether candidates should be promoted to FA or not [1]. Besides, any edit on Wikipedia articles could be viewed as being informally "peer reviewed" [37]. On most unprotected pages, any editors can "review" others' edits. If they don't like the edits, they can reject them by reverting or overwriting on them. In this paper, we focus on formal peer review systems where there are explicit roles like reviewers and submitters (which we refer as contributors in the following sections), and dedicated places to submit the review, explain the review decision and make arguments. Particularly, we examine what happens when submitter disagree with the reviewers (i.e., conflicts occur).

Effects on contributors' continuous participation

Researchers have found conflicts generally have negative effects on group loyalty, workgroup productivity, and job satisfaction [10, 19, 21]. For instance, Deutsch found that conflicts decrease goodwill and mutual understanding, which hinders the completion of organizational tasks [13]. Haq also suggested that conflict may increase employees' stress in workplace and lead to deviant behavior [19].

In the context of online collaborations, conflicts might cause contributors to feel that their individual goals no longer match the community goal and thus choose to leave the project. Furthermore, conflicts originally concerning the task itself can evolve into interpersonal or affective conflicts [6, 15]. This can lead to negative emotions such as feelings of disappointment, frustration, annoyance and anger, which also cause contributors to leave the community. Unlike traditional organizations, contributors of online collaboration projects have no formal employment or membership contacts with the community, so they can easily leave the communities with few social or economic consequences. Based on the above reasoning, we propose our first hypothesis:

H1: *Contributors who have conflicts with other members in the community are more likely to leave the community than those who don't have conflicts with other members*.

Strategies to Manage Conflict

Since conflicts can potentially cause serious issues and undermine the long-term success of the projects, it is important to handle and manage the conflicts. However, handling conflicts is not an easy job. Organizational conflict management theories suggest that conflict management involves "*designing effective macro-level strategies to minimize the dysfunctions of conflict and enhance the constructive functions of conflict in order to enhance learning and effectiveness in an organization*" [4]. Blake and Mouton first presented a conceptual scheme for classifying the modes for handling conflicts: forcing, withdrawing, smoothing, compromising, and problem solving [7].

However, despite the various strategies to manage conflict in traditional organizations, some of them might not be applicable in the context of online collaboration projects. For example, people's contributions are usually volunteer, which makes it difficult to use the "forcing" strategy to coerce any party of the conflict to reach consensus. To ensure the quality of productions, it is also not optimal to withdraw the conflict (i.e., ignore the issues) or compromise the existing standards or community goals. Thus, we focus our attention on the smoothing and problem solving strategies to manage conflicts in peer review process of online collaboration. Specifically, we identify three strategies to handle these conflicts: rational explanation, constructive suggestion and social encouragement. In the following section, we will present these strategies in detail and discuss their effects on contributors' continuous participation.

Rational Explanation

When a contributor disagrees with other members, a management strategy is giving him/her a rational explanation, which involves clarifying some potential misunderstandings, providing more details about the focal problems, and sometimes referencing other related resources. This strategy is aligned with the smoothing strategy in Blake and Mouton's classification [7].

The direct benefit of providing rational explanation is to minimize the misunderstanding. As many scholars on conflict negotiation, bargaining, mediation, and arbitration indicated [14, 27, 32], misunderstanding is one of the fundamental reasons of conflicts and disagreement. It is often useful to clarify the misunderstanding first. Another potential benefit of providing explanation is that members in the conflicts can learn some useful knowledge or skills about the production projects[28, 35], which might reduce the negative consequence of experiencing a conflict and keep them staying in the community. So we hypothesize:

H2-a: *Providing rational explanation to a contributor who has conflict with others can decrease the probability that s/he leave from the community.*

Constructive Suggestion

Providing constructive suggestions is corresponding to Blake and Mouton's problem solving strategy [7]. This strategy involves suggesting concrete alternative solutions to the conflict. Scholars have regarded problem-solving as one of the most common strategies to manage conflict. Rahim argued that, when dealing with conflicts, the first step was to diagnose the conflict and recognize the right problem, and then try best to find mutually acceptable solutions to the problem [4]. He proposed a regular process of problem solving: problem recognition, planning for change, and implementation.

The effectiveness of this strategy has been proven in traditional organization settings. Pruitt and Carnevale provided some evidence from laboratory studies indicating problem solving style is the best strategy in managing social conflict [30]. They found problem-solving was the most effective way to reach win-win outcomes, which can make both parties of the conflict satisfied. In the context of online collaboration projects, providing constructive suggestions can help contributors learn how to improve their contributions and know about directions of future work, which probably help them keep contributing to the communities.

H2-b: *Providing constructive suggestions to a contributor who has conflict with others can decrease the probability that s/he leave from the community.*

Social Encouragement

Social encouragement is another smoothing strategy. Unlike providing rational explanation which focuses on the issues, social encouragement aims to achieve friendly and social environment among members who are in conflicts. It is an attempt to build harmonious relationships between members by expressing grateful thanks, appreciating the enthusiastic work, and welcoming future contributions.

Although most conflicts in online collaboration are initially concerning the focal tasks, these conflicts can transform to affective conflicts which cause feelings of anger and frustration [6, 15]. Social encouragement can comfort contributors by providing emotional support and help build stronger cohesion between members. So, we predict:

H2-c: *Socially encouraging a contributor who has conflict with others can decrease the probability that s/he leave from the community.*

Conflict Manager's Administrative Role

In our study, another question of interest is whether the status of the person (e.g., with or without administrator roles) who delivered the conflict-management strategies influences the effectiveness of the strategies. For most online collaboration projects, there are some members holding the legitimate administrative roles (e.g., software projects' owners and managers, Q&A sites' moderators). They are responsible for setting the work directions and regulating members' behavior. The legitimate power of administrators usually stems from a selection process, such as being appointed from the projects' owners, or fulfilling some explicit criteria by long-term commitment [43].

Compared to regular members, administrators are more familiar with the projects, and usually hold higher authority and expertise in the community [29]. Prior research shows that administrators are generally more powerful in influencing and motivating others' activities [9]. For instance, Zhu et al. found that, in Wikipedia, the legitimate leaders were more influential in rewarding, regulating, directing and socializing other members compared to regular members [8]. Thus, the administrator's conflict-management behavior might also be more powerful than regular members. Therefore, we propose the following hypothesis:

H3: *The conflict-management strategies are more effective when they are delivered by members with administrator status.*

GITHUB AS STUDY PLATFORM

We choose GitHub as our study platform. It is a well-known online open collaboration platform [11], which is widely used to host open-source software projects. Through GitHub, a large number of developers can effectively collaborate to build open-source software. By the time of early 2015, GitHub has supported 9.1 million developers collaborating across 21.5 million code repositories [2]. GitHub has a peer review mechanism to ensure that any peer developers' code changes must be confirmed as valuable, secure and efficient before merging them into the code repositories. In this process, issues like competing technologies, incompatible software versions, and information overload can often cause conflicts [15]. Besides,

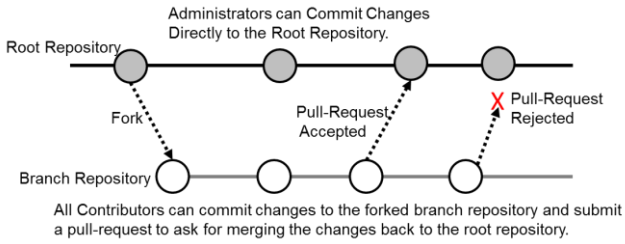


Figure 1. The fork-and-pull model of GitHub

most archival data of the hosted projects are publicly available through GitHub's official API [3]. Thus, we think GitHub is a suitable platform for our study.

To understand conflicts in GitHub, firstly, we need to learn about GitHub's *fork-and-pull* model and its peer review mechanism. As Figure 1 shows, GitHub implements a *fork-and-pull* model to support seamlessly exchanging commits across branches of the project. Each software project has a root code repository in GitHub. Only a relatively small set of contributors, who are usually the project's owners or some senior contributors (called "collaborators" in GitHub), have direct write permission to the root repository. These members with write permission to the root repository are administrators of the project. Those who don't have direct write permission can *fork* the root repository. This fork action makes a remote-hosted branch of the root repository. They can make changes on the branch repository or invite others to collaboratively work on it. If they want to contribute these changes back to the root repository, they can submit a "*pull-request*"(PR), asking for "*pulling*" their commits on the branch repository back to the root.

Then, there is a peer review process to decide whether the PR can be accepted. The reviewers can not only be the project's administrators but also other peer members. The reviewing is a highly interactive process, organized similar to a thread of online forum. The PR-contributor discusses with reviewers back and forth about the PR's usefulness and technical issues. Sometimes, reviewers might ask for further modifying the PR (e.g., write the proper documentation or add test-cases to validate the code). People can recognize whether a reviewer is the project's administrator from the label on the reviewer's comment. Only administrators can make the final decision about whether the PR should be accepted or not. Although sometimes the acceptance decision might be reversed, in this study we only consider the final decision.

What do Conflicts Look Like in GitHub?

In this section, we would like to use one example to show what conflicts look like in GitHub projects.

In a project developing a JavaScript application framework, a contributor submitted a PR named "*add 'set' event trigged once when collection models are added or removed*". He initially posted a comment to point out the performance issues of existing trigger method and proposed his approach to solve this problem. The exact code changes were also displayed. An owner of the project (R1) replied to it and said this functionality had been supported through the 'reset' method, "*Try using reset instead -- that's exactly what it's for*". However, the contributor firmly argued, "*I think reset (or fetch ({reset: true})) is okay, but I see 2 disadvantages compare to my approach…*". He outlined two points about why his approach is better.

The contributor's argument drew other members' attentions, and they joined the discussion to manage the conflict. Some offered suggestions for the focal problem. A member (R2) proposed another solution, "*Another solution to this is just to listen to the add/remove events, and throttle the rendering with underscore's throttle function. Simples?*". An administrator of the project (R3) replied, "*as an alternative you can trigger a 'set' event yourself* ", and he posted a code snippet to demonstrate an example solution to this problem. Some others chose to provide explanations about why this PR cannot be accepted. A member (R4) comprehensively explained that the DOM manipulation problem made this PR's approach was not feasible. In this progress, the PR-contributor actively discussed with all reviewers and still expressed disagreement with some reviewers' viewpoints. By the end, eighteen comments were posted for this PR and six reviewers joined the discussion, including two administrators and four regular members of the project. And the owner (R1) decided to reject this PR.

As we can see in the above example, conflicts in GitHub projects' peer review process are usually regarding the disagreement between contributors and reviewers on the acceptance of PRs. We use the PR-contributor's argument against reviewers as the signal of conflict's occurrence. And reviewers' conflict management strategies are identified from their responding comments to the argument. See Table 2 for comprehensive descriptions and examples of contributors' argument and reviewers' management strategies. In the present study, we will examine the effectiveness of these conflict management strategies on contributors' retention.

METHODS

Data Collection

We crawled our dataset through GitHub's official API [3] in March, 2015. As Kalliamvakou et al. pointed out, a large portion of GitHub's code repositories were for personal use [23]. Thus, to ensure the projects in our sample rely on a large number of volunteer developers' work, we randomly selected 170 projects that had been forked by at least 1000 users and had at least 100 PRs. The collected dataset mainly includes information about the projects' root repositories, pull-requests, and peer review comments. Among all the 196,037 PRs, 61.6% were accepted to be merged into the root repositories. The majority of projects (98.2%) were still receiving PRs during the last month before data collection. Table 1 shows the descriptive statistics of our dataset.

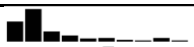
Variables	Median	Mean	Std. Dev.	Distribution	Max
Project's PRs	552.0	1153.2	1549.7		5661
Project's Percent of Accepted PRs	0.537	0.525	0.19		0.88
Project's Contributors	237.0	331.3	285.0		1778
Project's Administrators	6	10.7	14.3		52
Contributor's PRs in a Project	1.0	3.48	17.56		1092
PR's Contributor-Posting Comments	1.0	2.25	2.56		61
PR's Reviewer-Posting Comments	1.0	2.62	4.42		97
PR's Reviewers	1.0	1.46	1.62		80

Table 1. Descriptive statistics of the dataset

	Categories	Description and Examples
	Contributor's Argument	Contributor's disagreement on reviewers' criticism, defending their changes, or rejecting reviewer's proposals. "*I have to firmly disagree that `parse` is the best choice for this.*" "*… but I don't think there are any drawbacks compared to the mt_rand() method*"
Reviewer's Response	Rational Explanation	Explaining the problems in more details, clarifying some potential misunderstandings, giving extended knowledge or referencing other related resources. In the context of software development, the strategy is often aiming at explaining the reasons why the PR cannot be accepted. "*Your code was failing because even though you passed `unset: true`, the values are still equal.*" "*Oh you misunderstood what I meant. I didn't mean to question ...*"
	Constructive Suggestion	Providing concrete suggestions to solve the problem or guiding future work. In the context of software development, this strategy often provides suggestions to improve the existing solutions proposed by the pull request, providing alternative solutions, or providing concrete examples. "*How about something like this added to the CI superclass?... [Example Code]*" "*… you can just use the jQuery global error/success callbacks, provided for this purpose.*"
	Social Encouragement	Expressing grateful thanks, appreciating the enthusiastic work or creative ideas, and encouraging future contributions. "*… you are welcome to try your best at it - contributions are always welcome. :)*" "*thanks for your enthusiasm!*"

Table 2. Description and examples about contributors' argument and reviewers' three management strategies

Coding the Dataset

To identify PR-contributors' arguing comments and reviewers' management strategies, five coders initially hand-coded a sample of comments, then we trained classifiers to automatically code the whole dataset. The five coders are all graduate students in computer science and all had experiences in software programing.

The hand-coding process comprised two steps. First, we randomly selected 5000 PR-contributors' comments. Coders were instructed to label the comments which can indicate the PR-contributors' argument against reviewers. Coders were initially trained on a subset of 200 comments. The first author discussed with each of them to judge some ambiguous cases and understand the boundary between arguing and non-arguing comments. Then, they moved on to code the rest comments. Each comment was independently coded by two coders.

The next step is to code reviewers' responding comments into one or more categories of conflict-management strategy. We randomly selected 1000 PR-reviewers' comments posted behind the PR-contributors' arguing comments. Coders were instructed to identify whether the comments adopted one or more of the three conflict management strategies. The descriptions of these strategies, as Table 2 illustrates, were fully explained to them. These strategies are not mutually exclusive, which means that it is possible that a comment adopted multiple strategies.

The inter-judge agreements were evaluated using Cohen's Kappa [36]. All reached substantial or moderate level of agreement [1] (argument: 0.495, explanation: 0.678, suggestion: 0.529, social: 0.634). For those which can't be

[1] According to [36][35][38], it is a substantial level of agreement for kappa between 0.61 and 0.80, and a moderate level of agreement for kappa between 0.41 and 0.60.

judged consistently, coders discussed with each other and reached consistent judgements.

Next, we built classifiers to code the whole dataset. Specifically, two authors of this paper worked collaboratively to identify a set of features to represent the comment text. Then, the hand-coded dataset were randomly divided into training set (80%) and test set (20%). We implemented linear SVM model on the training set with 10-fold cross-validation, and evaluated the classification accuracy on the test set. [2] (See Appendix A. for details about the classifiers). The result shows that all classifiers achieved reasonably high accuracies (argument: 80.4%, explanation: 83.0%, suggestion: 79.1%, social: 87.3%). These classifiers were used to code the whole dataset. Firstly, all PR-contributor's arguing comments were found out (excluding comments which don't follow any reviewers' comments). Then, PR-reviewer's comments following behind the arguing comments were associated with particular conflict-management strategies. A comment might be coded with multiple strategies.

As a result, for 7.8% of PRs, their contributors had ever argued against reviewers. The explaining strategy is most frequently adopted to manage the disagreement (See Table 3). And, more than half of comments with management strategies were posted by administrators (See Figure 4).

	Frequency	Percent of being delivered by administrators
Explanation	51.2%	53.3%
Suggestion	29.2%	60.5%
Social	35.9%	80.1%

Table 3. Overview of the three strategies

ANALYSIS AND RESULTS

In this study, we applied survival analysis to test the hypotheses. Before that, we use propensity score matching (PSM) to reduce the bias introduced by pre-existing difference between contributors who argued against reviewers and those who did not argue.

Propensity Score Matching

In the context of our study, whether a contributor would argue against reviewers is not random. It is possible that the contributors who argued against reviewers are naturally different from those who did not argue. For example, the contributors who do not argue are more likely to be high-quality contributors or have similar mindset with others in the project. This pre-existing interpersonal difference can introduce bias in estimating the effects of conflict and its management strategies on the outcome (i.e., the likelihood of leaving the projects).

[2] We also tried some other text mining technologies, such as text classification methods (representing comments as a vector of words) and using LDA to model topics of comments. But the accuracy was poor.

This potential bias can be reduced using propensity score matching (PSM) approach. For each contributor who argued against reviewers, we selected a comparable contributor who did not argue but was most similar on a set of confounding variables, such as the prior experience of the contributor, the quality of his/her contributions (e.g., how many previous PRs made by him/her were rejected). See Appendix B for details about how we conducted PSM and the result shows bias is significantly reduced.

Survival Analysis

Next, survival analysis was applied to test the hypotheses of this paper. Survival analysis is a statistical technique to examine influences on time-related outcomes. In this study, our goal is to examine the effects of conflicts and its management strategies on the time of a contributor leaving the project (i.e., the time when a contributor stop submitting PRs to the project). We use survival analysis because standard regression models do not take into account *censored observations*. More specifically, during the observation period of this study, some contributors might stop submitting PRs for a short-term, but it is still possible that they would come back to the project and submit PRs again after the end time of this study (i.e., the time of data collection for this study). For these contributors, their time of leaving the project is unknown. In survival analysis, these censored observations are taken into account. So, it is more suitable for our study.

We adopted Cox proportional-hazards regression model for this study. This model is widely used in survival analysis. It assumes that the effects of predictive variables upon the hazard of event's occurrence are constant over time and are additive in one scale [26]. Since our predictive variables are time-dependent (e.g., the argument happened at a particular time, instead of being constant over time), we consider contributors' activities in a weekly scale. Specifically, the model treats a contributor's activities in a week as a record of the input data and examines the predictive variables' influences on whether the contributor would leave the project after the focal week [16]. As each record with argument has been matched with a record without argument in the PSM procedure, we grouped two matched records as a group and ran mixed-effect Cox regression to control the variance of different groups as random effects. The "*coxme*" function of an R-software package was used to run the analysis [38]. The Cox regression model's input and outcome variables are as follows.

Outcome Variable

LeaveProject: A dummy variable indicating whether a contributor left the project after the focal week (1 = leave, 0 = survive). We consider a contributor left the project if s/he does not submit any PR to the project after the focal week. But there is an exception. If a contributor does not submit PRs after the focal week but the focal week was within the last 20 weeks before the time of data collection, it is treated as censored observation.

Explanatory Variables

Argued: A dummy variable indicating whether the contributor argued against reviewers in the focal week. In our dataset, among all records that the contributor argued, most (95.8%) argued for only one PR.

Argued X Response: A dummy variable indicating whether reviewers responded to the contributor's argument. If a contributor argued for more than one PR in the focal week, we set this variable as true when any one of the PRs meets the conditions. So it is for the following variables.

Argued X ReByExplain: A dummy variable indicating whether reviewers responded to the contributor's argument by providing rational explanations.

Argued X ReBySuggest: A dummy variable indicating whether reviewers responded to the contributor's argument by providing constructive suggestions.

Argued X ReBySocial: A dummy variable indicating whether reviewers responded to the contributor's argument by giving social encouragements.

Argued X Admin: A dummy variable indicating whether administrators responded to the contributor's argument.

Argued X ReByExplain X Admin: A dummy variable indicating whether administrators responded to the contributor's argument by providing rational explanations.

Argued X ReBySuggest X Admin: A dummy variable indicating whether administrators responded to the contributor's argument by providing constructive suggestions.

Argued X ReBySocial X Admin: A dummy variable indicating whether administrators responded to the contributor's argument by giving social encouragements.

Control Variables

Three categories of confounding factors were included as control variables.

Contributors' confounding activities. The contributor's activities before and during the focal week are respectively calculated as control variables. Specifically, $PR\_Count_{<t}$ and $PR\_Count_{=t}$ are respectively the number of PRs a contributor submitted before the focal week (t) and during the focal week (t). $PostCommentsAvg_{<t}$ and $PostCommentsAvg_{=t}$ are respectively the average number of comments a contributor posted in a PR submitted before the focal week and during the focal week. $Rev\text{CommentsAvg}_{<t}$ and $RevCommentsAvg_{=t}$ are respectively the average number of comments a contributor received in a PR submitted before the focal week and during the focal week.

The quality of contributions. Three variables measuring the quality of a contributor's PRs are included as control variables. $RejectedPRPercent_{<t}$ and $RejectedPRPercent_{=t}$ are respectively the percent of rejected PRs among all PRs submitted before the focal week and during the focal week. *ArguedPR_Rejected* is a dummy variable indicating whether the PR for which the contributor argued was eventually rejected (1=rejected, 0=accepted).

Project-level variables. Three project-level variables are also included. *Project_Watchers* is the number of members who watched the project (the "watch" feature is similar to "following" a person in social network). *Project_Forks* measures how many times the project was forked. *Project_OpenIssues* is the number of publicly reported software issues.

All continuous variables were log transformed and standardized. The descriptive statistics in Table 5 are reported with values before log-transformation and standardization.

Results

Results of the models are reported in Table 5. All the four models were built using Cox Regression for modeling the effects of explanatory variables on the likelihood of leaving the project. The resulting effects are reported in the form of hazard ratio (HR), which is the predicted relative change in the probability of leaving the project caused by a unit increase in the predictive variable (e.g., a dummy variable changing from 0 to 1, or a continuous variable increasing by one unit of standard deviation).

Model 1 estimates the effect of conflicts (i.e., contributors' arguments against reviewers) on the probability of leaving the project, controlling the confounding factors. The result shows a significant effect. The hazard ratio value of 1.168 indicates that, if contributors argued against any reviewers in the focal week, they are 16.8% more likely to leave the project ($16.8\% = 1.168 \times 100\% - 100\%$, $p<0.001$).

So, H1 is supported.

Model 2 estimates the effects of reviewers' different conflict management strategies on the probability of the contributors' leaving the project. The result shows that, simply receiving replies has no significant effect on reducing the contributor's tendency of leaving (see *Argued X response*). Among the three types of conflict-management strategies, only providing constructive suggestions can significantly reduce the tendency of leaving the project. If reviewers responded to the contributors' argument by providing constructive suggestions, the contributor would be 21.7% less likely to leave the project ($-21.7\% = 0.783 \times 100\% - 100\%$, $p<0.01$). The other two strategies (i.e., providing rational explanations and social encouragements) have no significant effects on the contributor's tendency of leaving the project. Table 4 shows the correlation between the three strategies.

	Explanation	**Suggestion**	**Social**
Explanation		0.317	0.258
Suggestion			0.244

Table 4. Correlation between conflict management strategies

Variables	Descriptive Statistics		Model 1		Model 2		Model 3 (Excluding records that the PR with argument was accepted)		Model 4	
	Mean	Median	HR	SE	HR	SE	HR	**SE**	HR	SE
Control Variables										
$PR\_Count_{<t}$	7.79	3	0.304***	.062	0.302***	.062	0.338***	.085	0.302***	.062
$PR\_Count_{=t}$	2.49	1	0.646***	.033	0.651***	.033	0.659***	.044	0.650***	.033
$RevCommentsAvg_{<t}$	2.68	1.67	0.944	.056	0.946	.056	0.964	.074	0.946	.056
$RevCommentsAvg_{=t}$	7.79	5	1.096**	.024	1.115**	.026	1.078**	.032	1.114**	.026
$PostCommentsAvg_{<t}$	1.57	0.91	0.989	.049	0.987	.049	1.023	.065	0.986	.049
$PostCommentsAvg_{=t}$	4.65	3	0.736***	.026	0.745***	.026	0.766***	.032	0.746***	.026
$RejectedPRPercent_{<t}$	0.24	0.11	1.067**	.025	1.069**	.025	1.086*	.037	1.069**	.025
$RejectedPRPercent_{=t}$	0.50	0.50	1.129***	.017	1.103***	.021	1.091**	.031	1.103***	.021
ArguedPR_Rejected	0.24	0	/	/	1.149**	.056	/	/	1.147**	.056
Project_Watchers	6318	5399	1.085***	.017	1.088***	.017	1.112***	.023	1.088***	.017
Project_Forks	2195	1855	0.961**	.018	0.962**	.018	0.906***	.024	0.962**	.018
Project_OpenIssues	526	260	0.884***	.017	0.884***	.017	0.897***	.022	0.884***	.017
Explanatory Variables										
Argued	0.50	1	**1.168*** **	**.033**	1.112*	.076	1.131*	.087	1.112*	.076
Argued X Response	0.44	0			1.033	.074	1.089	.094	1.031	.102
Argued X ReByExplain	0.26	0			1.006	.053	1.068	.066	0.980	.084
Argued X ReBySuggest	0.15	0			**0.783****	**.057**	**0.822****	**.070**	0.829**	.091
Argued X ReBySocial	0.19	0			1.011	.050	0.945	.064	0.962	.095
Argued X Admin	0.31	0							1.001	.086
Argued X ReByExplain X Admin	0.18	0							1.037	.087
Argued X ReBySuggest X Admin	0.10	0							0.890	.103
Argued X ReBySocial X Admin	0.14	0							1.063	.101
Log-likelihood			-25957.3		-25942.2		-19305.2		-25941.4	
AIC			1336.8		1357.1		830.7		1350.6	
Chisq			1362		1393		874		1394	
Number of records			13732		13732		8560		13732	

***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$

Table 5. Cox proportional-hazards regression models predicting contributors' leaving from the projects.

One might speculate an intuitive explanation of the above results. The concrete suggestions to fix the PR's issues can help the PR-contributor to modify the PR. Thus, it is more likely that this PR is eventually accepted and conflict is probably resolved. But providing explanations or social encouragement has no such power. Given this reasoning, a question remains. When the PR is eventually rejected, is the suggesting strategy still effective to retain contributors? To examine it, we additionally built Model 3, which excludes all records that the PR with argument is accepted. The results remain the same. If reviewers responded to the argument by providing constructive suggestions and the PR was eventually rejected, the contributor would be 17.8% less likely to leave the project ($-17.8\% = 0.822 \times 100\% - 100\%$, $p<0.01$). Thus, we can conclude that constructive suggestions can effectively retain the contributors in conflict, whether the PR with argument is eventually accepted or not.

H2-b is supported, but H2-a and H2-c are not supported.

Model 4 considers whether the status of the people who delivered the management strategies would influence the effectiveness of the strategies. The result shows that whether the person responding to the argument is the project's administrator or not has no significant effect on the contributor's tendency to leave the project.

H3 is not supported.

SURVEY-BASED QUALITATIVE ANALYSIS

To further understand the quantitative results, we conducted an additional user survey. Many GitHub users left their email addresses on their GitHub profiles. Thus, based on our dataset, we found out the contributors who had ever argued against reviewers and collected their email addresses. An online survey was designed via Google Form and sent to these users. By the end, we received 16 respondents. 94.1% of them often or sometimes submitted PRs and all reported that they argued against reviewers at least once.

The survey mainly comprised open-end questions, so that we can collect the multifaceted viewpoints of respondents. Firstly, we asked whether the conflict with reviewers would make them less likely to keep submitting PRs in future. 81.3% of respondents gave the positive answer (i.e., "very possible" or "somewhat possible"). Reviewers' rude tone, unjustified criticism and disagreements on trivial issues are the main reasons leading to leaving from projects.

In the following questions, we presented the three types of reviewers' responses to manage the disagreement and asked respondents whether these types of responses can eliminate

their dissatisfaction and make them more likely to keep submitting PRs. (See Appendix-C for details)

For the first type of response, that is explanation of why rejecting the PR, P7 pointed out that,

"*These kind of responses (explanations) are only positive when they are "for real", ie, pointing real problems/issues, with clarity. Unfortunately, it is not uncommon to see replies, usually from supporters (not core/leaders), which fail to understand your work and instead to collaborate on the issue only create noise and distraction.*"

This quote identifies a potential reason for why the explaining strategy is not effective in retaining contributors. Although the explanations come with good intent, the explanations might be superficial. Instead of making problems/issues clear, the explanations actually make the "misunderstanding" clearer and show the contributors how ignorant the reviewers are about their PRs, which frustrates the contributors. In contrast, the concrete suggestions are often based on thorough understanding of the contributors' work, which provides a signal to contributors that their original work or ideas are appreciated. The use of explanation strategy or suggestion strategy might indicate the different levels of efforts the reviewers and others make to understand the contributors' work, which might be the underlying drive that causes contributors to leave or stay.

For social encouragement, several respondents explicitly shared their viewpoints about why this type of responses is not always welcomed. P13 believed that these social messages sound abnormal. P13 said "*explicitly expressing these attitudes runs the risk of sounding overly polite/verbose to me*". When dealing with conflict, the normal communications are concise and objective. Sugarcoating the conflicts by adding social encouragement runs the risk of deviating this norm.

Furthermore, P14 suggested that contributors in software development projects might have their own code to interpret the social encouragement. P14 made an analogy to the scenario of receiving a testimonial from a former employer,

"*When you get a testimonial for an employment you once had there are some codes (at least in the country I live). 'He has always been giving his best' is like saying: 'don't even consider to recruit this guy'. You don't thank for 'enthusiasm' or encourage to 'try your best'.*"

This quote indicates that, when dealing with conflicts, the social encouragements might be interpreted as just a way to say that their work are not good enough to be accepted, which is not comforting at all.

P13 also said he would feel more negative if he found these responses are automated, "*if I ever get the feeling that such responses are in any way automated, I'll completely dismiss them and similar responses in the future and just take them as 'marketing'*". The comment serves a warning if any online collaboration system tries to create any automatic-generated review to deal with conflict. P2 pointed out that the social strategy might only be useful for newcomers, "*These comments aren't very useful to me. But I imagine they could be useful for new contributors who don't feel very confident.*"

DISCUSSION

This paper mainly explored the plausible strategies to manage the conflicts which arise in the peer review process of GitHub projects. We identified three conflict-management strategies and quantitatively examined their respective effectiveness in retaining contributors in the projects.

The quantitative results show that providing contributive suggestions is effective in reducing the likelihood of contributors' leaving from the projects, while pure explanations and social encouragement have no effects in retaining the contributors. Further qualitative survey suggests that the reason why explanations are not effective might be that the explanations provided in the wild are often superficial and misunderstand the contributors' work. Since contributors in these projects are volunteers, intrinsic motivations such as feeling competent and being understood are important factors to keep them working [34]. The explanation strategy is not effective because it often fails to meet the contributors' intrinsic needs.

The survey also provides an explanation about why social encouragement fails. It is possible that social messages might be interpreted as a way to say one's work is not good enough to be accepted. Note that prior research found positive effects of social messages on receivers' subsequent participation in other open collaboration communities. In Zhu et al.'s paper [44], they found that person-based messages (which are similar to the social encouragement strategy in this paper) significantly increase the receivers' subsequent contributions by 23%. It is possible that messages with warm and social tones are more encouraging in generally motivating members' engagement than in dealing with conflicts. Therefore, the usage context might play an important role in deciding whether the social strategy would work.

Our paper built hypotheses based on theories from traditional conflict management developed in offline organizations. Our empirical results show that some hold in the online collaboration settings, but others do not. The differences enhance our understanding of the online open collaborations.

Practical Implications for Conflict Management

Although the present study focuses on the conflicts which arise in the peer review process of GitHub projects, we believe our findings have implications for wide range of online collaboration systems. For instance, as Stvilia et al. argues [37], Wikipedia's open editing model can be viewed as an informal peer review mechanism where all

contributions are initially accepted and then other editors perform review and reject unwanted contributions. When editing conflict occurs, editors can collectively discuss the conflict on the article's talk page [25]. This scenario is quite similar to the context of our study. And we believe our work provide some new insights about the effective strategies to manage these conflicts.

Our results confirm that conflicts cause members to leave the project. So, it is important to design tools to promote the awareness of conflicts. This paper shows that, through training classifiers, conflicts can be automatically detected from members' communication content (e.g., peer review comments) with reasonably high accuracy. Based on these models, we suggest creating tools like "conflict dashboard", which can visually display the ongoing conflicts in a project. These tools can help community members be timely aware of the presence of conflicts and call for necessary management.

Although prior studies developed some conflict management procedures or policies (e.g., [12, 24]), they rarely suggest which way to communicate with members in conflict is more effective. Our results show that, compared to giving explanations or social encouragement, providing the concrete suggestions is the most effective way to manage conflict. Based on this finding, we suggest existing communities should educate members to try to understand the intent of the contributors and provide constructive suggestions when conflicts occur. The online collaboration systems should also support flexible interfaces for members to conveniently provide suggestions. In Wikipedia, newcomers' edits are often reverted without any further information. According to Halfaker's paper[17], this might cause the slower growth of Wikipedia editor base. They then proposed a tool to present newcomers' activity traces to Wikipedian mentors, and these information significantly helped mentors to provide more explanations about edit reverts [18]. However, our results suggest that tools should be designed to support not just providing explanations but also giving constructive suggestions for solving the problem in order to retain good-faith contributors.

Limitations and Future Research

Although we have used propensity score matching to reduce the bias caused by pre-existing interpersonal differences between contributors, we still cannot fully control all the confounding factors. For example, we don't have good measurement for the quality of each specific pull request. It is possible that people tend to provide explicit improvement suggestions for relatively high quality pull request but only provide explanations or social encouragement for relatively low quality pull request. We have included the variable (*ArguedPR_Rejected*) measuring whether the pull request was eventually accepted or not as control variables, but this dummy variable might not be accurate enough for measuring the quality of the pull requests.

Secondly, we did not distinguish the different nature of conflict. Organizational literatures suggested that the choice of conflict management strategy might depend on the type of the conflicts [4]. For example, when the conflicts are originated by minor issues but are turned into affective conflicts with harsh criticism, social encouragement might be appropriate. We suggest future research can conduct sentiment analysis on reviewers' criticisms and contributors' arguments. By this way, conflicts with harsh criticism or affective complain can be differentiated from other conflicts. Then people can investigate effective strategies to manage these different types of conflict.

Since conflict is a heterogeneous and multifaceted phenomenon, the conflicts studied in this paper cannot represent some other conflict scenarios in online collaboration contexts. We encourage researchers to explore the patterns of conflict management in wider range of online collaboration systems. And researchers can also investigate the influences of conflict and its management on other outcomes (e.g., the influences on reviewers' experience).

CONCLUSION

This paper identified three strategies to manage conflicts in peer review process of online collaboration projects and quantitatively examined their respective effectiveness in retaining contributors in the projects. The result shows that neither rational explanations nor social encouragement could reduce the likelihood of contributors' leaving from projects. Only providing constructive suggestions can effectively retain the contributors. And the status of the person who delivered the conflict management strategies has no significant influence on the contributor's continuous participation.

REFERENCES

1. Wikipedia:Featured Article Candidates.Retrieved July 20, 2015 from https://en.wikipedia.org/wiki/Wikipedia:Featured_article_candidates
2. Press ·GitHub.Retrieved July 20, 2015 from https://github.com/about/press
3. GitHub Developer.Retrieved July 20, 2015 from https://developer.github.com/
4. M Afzalur Rahim. 2002.Toward a Theory of Managing Organizational Conflict. *International Journal Of Conflict Management* 13, 3: 206-235.
5. Allen C. Amason.1996.Distinguishing the Effects of Functional and Dysfunctional Conflict On Strategic Decision Making: Resolving a Paradox for Top Management Teams. *Academy Of Management Journal* 39, 1: 123-148.
6. Ofer Arazy and Lisa Yeo et al. 2013. Stay On the Wikipedia Task: When Task-Related Disagreements Slip Into Personal and Procedural Conflicts. *Journal of*

the American Society for Information Science and Technology 64, 8: 1634-1648.
7. Robert R. Blake and Jane S. Mouton, et al.1962.Managerial Grid. *Advanced Management - Office Executive* 1, 9: 12-15.
8. Marco Caliendo and Sabine Kopeinig. 2008. Some Practical Guidance for the Implementation of Propensity Score Matching. *Journal Of Economic Surveys* 22, 1: 31-72.
9. Robert B. Cialdini and Noah J. Goldstein.2004. Social Influence: Compliance and Conformity. *Annual Review of Psychology* 55: 591-621.
10. Richard A. Cosier and Gerald L. Rose.1977.Cognitive Conflict and Goal Conflict Effects On Task Performance. *Organizational Behavior and Human Performance* 19, 2: 378-391.
11. Laura Dabbish and Colleen Stuart, et al.2012.Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository. *In Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 1277-1286.
12. Juliana de Melo Bezerra and Celso Massaki Hirata.2012.Applying Conflict Management Process to Wiki Communities. In *Enterprise Information Systems*. Springer, 333-348.
13. Morton Deutsch.1969.Conflicts: Productive and Destructive*. *Journal Of Social Issues* 25, 1: 7-42.
14. Morton Deutsch and Perter T. Coleman et al.2011.*The Handbook of Conflict Resolution: Theory and Practice*. John Wiley & Sons.
15. Anna Filippova and Hichang Cho.2015.Mudslinging and Manners: Unpacking Conflict in Free and Open Source Software.*In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 1393-1403.
16. John Fox.2002.Cox Proportional-Hazards Regression for Survival Data. *An R and S-PLUS companion to applied regression*: 1-18.
17. Aaron Halfaker and R Stuart Geiger, et al.2012.The Rise and Decline of an Open Collaboration System: How Wikipedia’S Reaction to Popularity is Causing its Decline. *American Behavioral Scientist*: 1746469259.
18. Aaron Halfaker and R Stuart Geiger, et al.2014.Snuggle: Designing for Efficient Socialization and Ideological Critique. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 311-320.
19. Haq, I.U.2011.The Impact of Interpersonal Conflict On Job Outcomes: Mediating Role of Perception of Organizational Politics.*Procedia-Social and Behavioral Sciences* 25: 287-310.
20. Pamela J. Hinds and Diane E. Bailey.2003.Out of Sight, Out of Sync: Understanding Conflict in Distributed Teams. *Organization Science* 14, 6: 615-632.
21. Karen A. Jehn.1997. A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups. *Administrative Science Quarterly*: 530-557.
22. Boris Kabanoff.1985. Potential Influence Structures as Sources of Interpersonal Conflict in Groups and Organizations. *Organizational Behavior And Human Decision Processes* 36, 1: 113-141.
23. Eirini Kalliamvakou and Georgios Gousios, et al.2014.The Promises and Perils of Mining GitHub.*In Proceedings of the 11th Working Conference on Mining Software Repositories*, 92-101.
24. Kittur, A. and Kraut, R.E.2010.Beyond Wikipedia: Coordination and Conflict in Online Production Groups.*In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, 215-224.
25. Kittur, A. and Suh, B., et al.2007.He Says, She Says: Conflict and Coordination in Wikipedia.*In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 453-462.
26. Kleinbaum, D.G. and Klein, M.1996.*Survival Analysis*. Springer.
27. Lewicki, R.J. and Weiss, S.E., et al.1992.Models of Conflict, Negotiation and Third Party Intervention: A Review and Synthesis.*Journal Of Organizational Behavior* 13, 3: 209-252.
28. Luthans, F. and Rubach, M.J., et al.1995.Going Beyond Total Quality: The Characteristics, Techniques, and Measures of Learning Organizations.*The International Journal of Organizational Analysis* 3, 1: 24-44.
29. O'Mahony, S.A.N. and Ferraro, F.2007.The Emergence of Governance in an Open Source Community.*Academy Of Management Journal* 50, 5: 1079-1106.
30. Pruitt, D.G. and Carnevale, P.J.1993.*Negotiation in Social Conflict*. Thomson Brooks/Cole Publishing Co.
31. Rahim, A. and Bonoma, T.V.1979.Managing Organizational Conflict: A Model for Diagnosis and Intervention.*Psychological Reports* 44, 3c: 1323-1344.
32. Ramsbotham, O. and Miall, H., et al.2011.*Contemporary Conflict Resolution*. Polity.
33. Rigby, P.C. and German, D.M., et al.2014.Peer Review on Open-Source Software Projects: Parameters, Statistical Models, and Theory.*ACM Transactions on Software Engineering and Methodology (TOSEM)* 23, 4: 35.
34. Ryan, R.M. and Deci, E.L.2000.Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being.*American Psychologist* 55, 1: 68.
35. Senge, P.M.2006.*The Fifth Discipline: The Art and Practice of the Learning Organization*. Broadway Business.
36. Stemler, S.2001.An Overview of Content Analysis. Practical Assessment Research Evaluation. Xu and Zhang.(2005).*International migration of nurses: Political and policy issues and Implications. Retrieved December* 10: 2009.
37. Stvilia, B. and Twidale, M.B., et al.2008.Information Quality Work Organization in Wikipedia.*Journal of the American Society for Information Science and Technology* 59, 6: 983-1001.
38. Therneau, T.2012.Mixed Effects Cox Models.*Mayo*

Clinic, Mayo Foundation for Medical Education and Research, http://cran. r-project. org/web/packages/coxme/vignettes/coxme. pdf.
39. Thompson, L.L. and Mannix, E.A., et al.1988.Group Negotiation: Effects of Decision Rule, Agenda, and Aspiration.*Journal Of Personality And Social Psychology* 54, 1: 86.
40. Tsay, J. and Dabbish, L., et al.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*, 144-154.
41. van Wendel De Joode, R.2004.Managing Conflicts in Open Source Communities.*Electronic Markets* 14, 2: 104-113.
42. Wang, J. and Shih, P.C., et al.2015.Revisiting Linus' S Law: Benefits and Challenges of Open Source Software Peer Review.*International Journal of Human-Computer Studies* 77: 52-65.
43. Yukl, G.A.1994.*Leadership in Organizations*. Pearson Education India.
44. Zhu, H. and Kraut, R., et al.2012.Effectiveness of Shared Leadership in Online Communities.*In Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 407-416.

APPENDIX

A. Training the Classifiers

We built four classifiers in total. The first classifier aims to classify whether a PR-contributor-posting comment is an arguing comment. Besides some explicit disagreement or argument words, some linguistic patterns also have strong predictive power. For example, **<*SUPPORT+BUT*>** represents saying some supportive or agreeing words at first but then expressed the opposite opinions, e.g., "understand … but …", "agree … but … ". **<*BUT+NEGATION*>** and **<*BUT+MORE*>** represent opposite opinions containing negations or comparative words, such as "but…ugly…", "however…insecure…", "but … more elegant …". When predicting whether a PR-reviewer's comment contains explanations, ***sentence count*** and ***word count per sentence*** and some reasoning words (e.g., ***because***, ***reason***) have most significant predictive power. The ***question*** feature **(**i.e., the number of questions**)** contributes as a negative predictor. ***Code Snippets*** and **<*You+MODAL*>** (e.g., "you can", "you should") are two most significant predictors for suggesting comments. The socially encouraging comments can be predicted by some appreciating or supportive words, such as ***thank, awesome, good points, welcome.***

B. Propensity Score Matching

The first step of PSM is to estimate the propensity score. In this study, the propensity score is the probability that a contributor would argue against reviewers in a particular week (*t*). Based on a logistic regression model, we estimate it from a set of covariates. Table 6 shows the result of the logistic regression.

Variables	Coef.	Std. Err.
Intercept	-.1041***	.0019
$PR\_Count_{<t}$	-.0007	.0005
$PR\_Count_{=t}$	.0825***	.0019
$RejectedPRPercent_{<t}$	-.0006	.0033
$RejectedPRPercent_{=t}$	.0079 ***	.0019
$RevCommentsAvg_{<t}$	-.0096***	.0020
$RevCommentsAvg_{=t}$	.0299***	.0012
$PostCommentsAvg_{<t}$	.0154***	.0024
$PostCommentsAvg_{=t}$	.1027***	.0014

***: $p < 0.001$, AIC = -31262, N = 121,448

Table 6. Logistic regression estimating the probability (propensity score) that a contributor would argue against the reviewer in the focal week. [3]

Then we matched each contributor who argued against reviewers in the focal week with a contributor who satisfied all the following four conditions: (i) s/he had ever submitted PRs to the same project; (ii) s/he did not argue against any reviewer in the focal week; (iii) s/he had **closest** propensity score; (iiii) s/he had not been previously selected to match with another arguing contributor. As Table 7 shows, before matching, the differences between the arguing contributors (treatment groups, N=6866) and the non-arguing contributors (control group, N=114,582) on all covariates are very significant (most covariates' biases are over 30%[4]). After matching, each unit in the treatment

Variables		Before Matching		After Matching	
		Mean	Bias	Mean	Bias
$PR\_Count_{<t}$	Treat	39.36	25%	39.3	0.3%
	Ctrl	18.64		39.0	
$PR\_Count_{=t}$	Treat	2.57	35%	2.57	4%
	Ctrl	1.56		2.42	
$RejectedPRPercent_{<t}$	Treat	0.24	16%	0.24	1%
	Ctrl	0.19		0.24	
$RejectedPRPercent_{=t}$	Treat	0.51	17%	0.51	2%
	Ctrl	0.43		0.50	
$RevCommentsAvg_{<t}$	Treat	2.67	34%	2.67	0%
	Ctrl	1.46		2.68	
$RevCommentsAvg_{=t}$	Treat	8.10	71%	8.10	6%
	Ctrl	2.43		7.49	
$PostCommentsAvg_{<t}$	Treat	1.56	39%	1.56	1%
	Ctrl	0.74		1.59	
$PostCommentsAvg_{=t}$	Treat	4.88	86%	4.88	8%
	Ctrl	1.04		4.42	

Table 7. Comparing the treatment group (i.e., arguing contributors) and the control group (i.e., non-arguing contributors) on the covariates before matching and after matching.

[3] The covariates used in the logistic regression model have been explained in the "Control Variable" sub-section (See page 7). All independent variable were log transformed.

[4] Bias is an indicator to assess the difference between the treatment group and control group on a covariate. It is

group has exactly one matched unit in the control group. And the differences are greatly reduced (after matching most covariates' biases are less than 5%). In such a way, the treatment group and control group are well balanced.

C. Open-end Survey Questions

Imagine that you submitted a pull-request and disagreed with your reviewer(s). You felt frustrated and seriously thought about quitting the project. You expressed your disagreement by saying things like:
"I don't think that solution works more nicely."
"I have to firmly disagree that `parse` is the best choice for this."
In such scenario, what kinds of responses described below is most helpful in resolving the disagreement?

Response Type 1: Providing explanation of rejecting your work
Specifically, reviewers would give you some detailed explanations about why your pull-request cannot be accepted or clarify some potential misunderstandings.
Example 1: *"Your code was failing because even though you passed `unset: true`, the values are still equal."*
Example 2: *"Oh you misunderstood what I meant. I didn't mean to question the usefulness of your PR."*
Can this kind of responses eliminate your dissatisfaction, frustration, or any other negative feelings? In addition, if you received this kind of responses, are you more likely to keep submitting pull-requests to the repository? Tell us why.

Response Type 2: Providing Constructive Suggestions
Specifically, reviewers would provide concrete suggestions to improve the existing solutions, or propose some alternative solutions.
Example 1: *"How about something like this added to the CI superclass?... [Example Code]"*
Example 2: *"... you can just use the jQuery global error/success callbacks, provided for this purpose."*
Can this kind of responses eliminate your dissatisfaction, frustration, or any other negative feelings? In addition, if you received this kind of responses, are you more likely to keep submitting pull-requests to the repository? Tell us why.

Response Type 3: Giving Social Encouragement
Specifically, reviewers would express grateful thanks, appreciate your enthusiastic work, or encourage future contributions.
Example 1: *"... you are welcome to try your best at it - contributions are always welcome. :)"*
Example 2: *"thanks for your enthusiasm!"*
Can this kind of responses eliminate your dissatisfaction, frustration, or any other negative feelings? In addition, if you received this kind of responses, are you more likely to keep submitting pull-requests to the respository? Tell us why.

---

defined as $\left|\overline{X_t} - \overline{X_c}\right| / \sqrt{(S_t^2 + S_c^2)/2}$, where $\overline{X_t}$ and $\overline{X_c}$ are the sample means, S_t^2 and S_c^2 are the sample variances[8].