This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

How Do Users Revise Answers on Technical Q&A Websites? A Case Study on Stack Overflow

Shaowei Wang, *Member, IEEE* Tse-Hsun (Peter) Chen, *Member, IEEE* and Ahmed E. Hassan, *Member, IEEE*

Abstract—To ensure the quality of its shared knowledge, Stack Overflow encourages users to revise answers through a badge system, which is based on quantitative measures (e.g., a badge is awarded after revising more than 500 answers). Prior studies show that badges can positively steer the user behavior on Stack Overflow (e.g., increasing user participation). However, little is known whether revision-related badges have a negative impact on the quality of revisions since some studies show that certain users may game incentive systems to gain rewards. In this study, we analyze 3,871,966 revision records that are collected from 2,377,692 Stack Overflow answers. We find that: 1) Users performed a much larger than usual revisions on the badge-awarding days compared to normal days; 25% of the users did not make any more revisions once they received their first revision-related badge. 2) Performing more revisions than usual in a single day increased the likelihood of such revisions being rolled back (e.g., due to undesired or incorrect revisions). 3) Users were more likely to perform text and small revisions if they performed many revisions in a single day. Our findings are concurred by the Stack Overflow community, and they highlight the need for changes to the current badge system in order to provide a better balance between the quality and quantity of revisions.

Index Terms—Stack Overflow, Incentive System, Badge, Answer Revision

1 Introduction

Technical question and answer (Q&A) websites have changed how developers seek information on the web. Q&A websites are becoming an important and popular platform for knowledge sharing and learning. When facing problems, users often seek help from other developers by posting questions on these Q&A websites (e.g., Stack Overflow[1], Quora[2], and MSDN forum[3]). As an example, Stack Overflow, one of the most popular Q&A websites for developers, has more than 16 million questions, 25 million answers, 68 million comments, and 50 million monthly visitors as of September 2018[4].

However, asking and answering questions on Q&A websites may not always be straightforward. For instance, answers may lack the explanations for some important concepts or references, which may in turn make such answers difficult to understand [1]. In addition, answers may contain incorrect information or buggy code snippets. Hence, one significant challenge for Q&A websites is ensuring the quality of their content [2].

- *S. Wang and A. E. Hassan with the School of the Software Analysis and Intelligence Lab (SAIL) in the School of Computing at Queens University, Canada.*
  E-mail: {shaowei, ahmed}@cs.queensu.ca
- *T-H. Chen is with the Department of Computer Science and Software Engineering at Concordia University, Montreal, Canada.*
  E-mail: peterc@encs.concordia.ca

1. https://stackoverflow.com/
2. https://www.quora.com/
3. https://social.msdn.microsoft.com/Forums/en-US/home
4. https://insights.stackoverflow.com/survey/2018

As a result, Q&A websites have developed several mechanisms to ensure the quality of their content (e.g., reviewing of questions and revisions, as well as revising questions and answers). A major mechanism on Stack Overflow to encourage users to revise answers is the use of a badge system. Users are awarded badges based on quantitative measures (e.g., by revising more than 500 answers on Stack Overflow). Such badges aim to encourage the positive contributions (e.g., improving the quality of content) of users on Q&A websites. However, such revision-related badges only consider the quantitative measures of revisions and not their quality.

Prior studies show that badge systems can positively steer user behaviors on Q&A websites, e.g., [3], [4] observed that a badge can increase the overall level of user participation. On the other hand, some prior studies show that incentive systems may not always drive certain users in a positive way on Q&A websites [5], [6], e.g., users may aggressively game the system for profit.

Therefore, in this study, we wish to investigate how the badge system steers the revision behavior of users on Stack Overflow. For example, do users' revision activities change as they are about to receive badges? We are also interested in investigating the potential threats of such user behavior changes on the quality of revisions. For example, does making more revisions in a single day decrease the quality of revisions (e.g., increasing the likelihood of such revisions being roll backed)? A better understanding of the badge rewarding may help Q&A website designers improve the current badge system (e.g., reworking some badges or creating new ones to reduce the number of low-quality revisions).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

To understand the efficacy of the badge system and whether encouraging some revision activities may have a negative impact on the quality of revisions, we studied 3,871,966 answer revisions based on the 2,377,692 answers that were created on Stack Overflow from August 2008 to March 2017. These answers involved 280,617 unique users. We study the collected data to understand how users revise answers over time. More specifically, we studied the revision patterns of users (e.g., do users revise answers in bursts or infrequently over a long period of time), especially when they are about to receive revision-related badges. Moreover, examining the content of revisions can help us better understand the efficacy of revision badges. Therefore, we investigated which parts (i.e., text block or code block) of the answers revised, as well as the underlying reasons for such revisions. We also examined rolled back revisions and the factors that are associated with such rollbacks. In particular, we address the following research questions:

- **RQ1: Do badges change the number of user revisions over time as they are about to receive a badge?**
  Users performed considerably more revisions (statistically significant) on the badge-awarding days than ones on normal days. In addition, users that were awarded revision-related badges (i.e., badge-holders) were 17 times more likely to perform spikes of revisions (i.e., perform a larger than usual number of revisions on a certain day) than those that were never awarded revision-related badges (i.e., non-badge-holders). Finally, 25% of the users did not make any more revisions after they received their first revision-related badge.
- **RQ2: How do users revise answers in terms of their content and size?**
  Users were more likely to perform small and text revisions when they perform many revisions in a single day. Users are more likely to perform simpler revisions (i.e., Text Correction and Code Formatting) on other users' answers.
- **RQ3: Which factors are associated with answer rollbacks?**
  Making a spike of revisions in a single day increases the likelihood of a revision being rolled back (i.e., rejected by earlier answerers). Through a qualitative study, we observed that answer revisions were often rolled back due to undesired code/text formatting, incorrect code revisions, and other text-related revisions.

In short, we observed that users usually perform more text and minor revisions when they perform more revisions in a single day (e.g., when pursuing badges). However, making more revisions than usual in a single day increases the likelihood of a revision being rolled back due to making undesired/incorrect revisions.

Based on our findings, Stack Overflow website designers may wish to adjust their current badge system to improve the answer revision process. For instance, Stack Overflow website designers may consider to incorporate certain rules to create a better balance between the quality and quantity of revisions. Stack Overflow designers may consider changing their badges to only consider unrolled-backed revisions instead of simply counting all revisions. We discussed our findings with the Stack Overflow community and several community members concurred that our findings highlight the need for an improvement to some of their current badges. Our study highlights the need for future research to devise ways to improve the revision-related badges.

The structure of this paper is as follows. Section 2 introduces the background about Stack Overflow, the answer revision process, and the current badge system. Section 3 describes our research questions and our data collection process. Section 4 presents the results of our research questions. Section 5 discusses the feedback that we collected from the Stack Overflow Meta forum[5] and the implications of our study. Section 6 presents the threats to validity of our study. Section 7 overviews the related work. Finally, Section 8 concludes our study and discusses possible follow-up studies.

2 Background

In this section, we give a brief overview of Stack Overflow, how users revise their answers, and the current badge system of Stack Overflow by using actual examples from Stack Overflow.

2.1 The Question Asking and Answering Process on Stack Overflow

Stack Overflow allows users to register, post questions, answer questions, leave comments on posts (i.e., either questions or answers), revise posts, vote on posts, and search or browse posts based on tags. Users can include code snippets and other references (e.g., URLs or images) to enrich their questions. Note that we define code snippets as *code block* in the rest of the paper and all the other non-code content (e.g., textual description, URLs) as *text block*. Other users could answer posted questions based on their experience. Each question may receive multiple answers from different users. However, at most one answer could be marked as accepted by the user who initially asked the question (to indicate that this particular answer is the most suitable/correct one). The scores of a post (i.e., either a question or an answer) indicate the total number of up and down votes that this post has received. Figure 1 shows an example snapshot of a question and its corresponding answers.

2.2 Improving the Quality of Questions and Answers on Stack Overflow

Stack Overflow encourages users to improve the quality of posts through two mechanisms: 1) encouraging users to revise posts (i.e., revision process); 2) encouraging users to review posts and revisions (i.e., review process). However, revising and reviewing are two different processes. For example, Stack Overflow only allows users with more than 2,000 reputation scores to do reviewing, while Stack Overflow encourages every user to perform revisions.

An overview of the answer review process is presented in Figure 2. Users can review the posts that are displayed on

5. https://stackoverflow.com/help/whats-meta

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

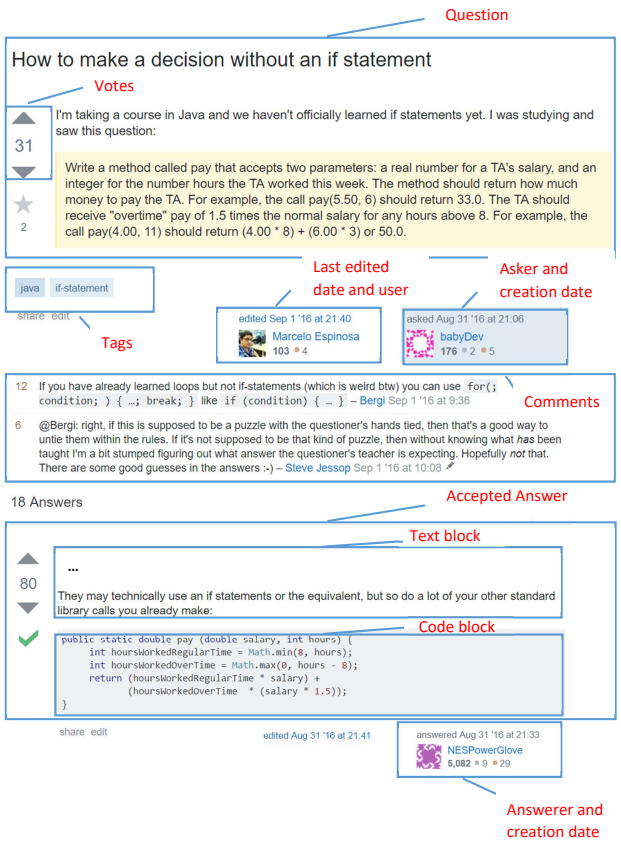

Fig. 1: An example question and its accepted answer on Stack Overflow. The example highlights many details that one can observe such as: a user asked "how to make a decision without an if statement" on August 31, 2016 and received an answer from another user. The answer was then accepted by the asker. The answer received 80 scores from the community. The question is associated with tags "java" and "if-statement".

the website or that are in the review queues (e.g., a queue for low-quality posts) and perform a revision on a post. Once a revision is performed by a user, the revision will be added into a suggested revision review queue and wait for users with more than 2,000 reputation scores to review it. If the revision is performed by a user with more than 2,000 reputation scores, the revision will be applied to the answer without any review process.

Among all the answer revisions, we find that only 9% of them went through the review process. In other words, 91% of the answer revisions were made by users with more than 2,000 reputation scores, and thus, these revisions were not assured by the revision review process. Therefore, in this study, we mainly focus on studying the answer revision process, which is the main quality assurance process for answers on Stack Overflow. Figure 3 shows an example of a user revising the content of an answer to make the answer more accurate and concise.

Figure 4 shows an example of a rollback[6]. The person who performed the rollback mentioned why he rolled back the answer: "when [other users] edited this answer and added some syntax highlighting it became incorrect and no longer made sense, apparently they didn't understand the material". Through such an example, we find that although the main purpose of revising an answer is to improve the answer quality, sometimes the revisions may be undesirable or even lead to an incorrect answer. Hence, if such incorrect revisions were not rolled back, such revised answers would mislead other users on Stack Overflow. As more of such undesired/incorrect revisions are performed, there is a higher likelihood of some of them remaining and not getting rolled back. Hence, it is important for Stack Overflow to reduce or avoid such cases. In fact, Stack Overflow requires the reviewing of a revision that is performed by a user with less than 2,000 reputation scores before it is applied to the corresponding answer. However, we still observe a notable number of rolled back revisions in Stack Overflow. Thus, we want to explore the reasons that the revisions are rolled back.

6. http://stackoverflow.com/posts/7266617/revisions

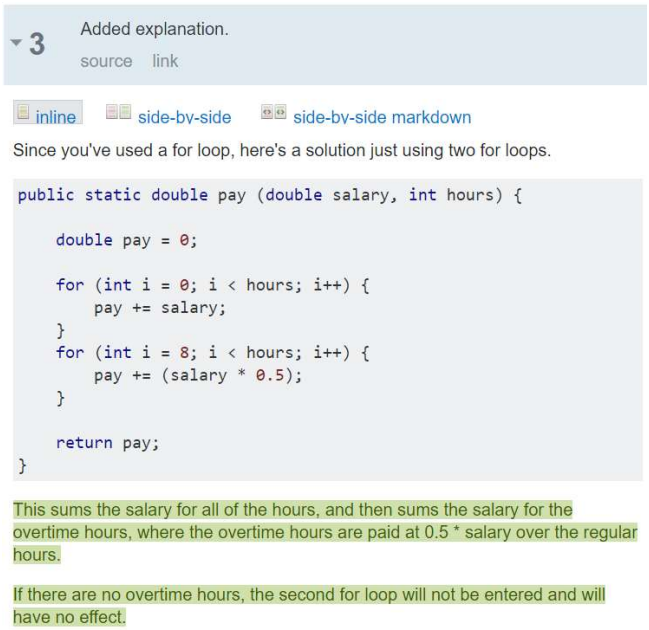

Fig. 3: An example of an answer revision (the added explanations is highlighted in green by Stack Overflow).

Stack Overflow provides a platform for users to search for answers. Having a clear and correct answer is very important for such answer seekers. However, we find that the revisions that are made on answers do not always improve the understandability of an answer, and in some instances the revisions may even be incorrect (Figure 4 provides such an example). We also compared the frequency of revisions that were made on questions and answers and found that revisions are performed more frequently on answers (1.6 revisions per answer) than on questions (1 revision per question). Thus, in this paper, we focus our study on how users revise answers and the reasons that cause such revisions to be rolled back. In short, this paper studies the efficacy of the badge system and offer insights into improving the answer revision process.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

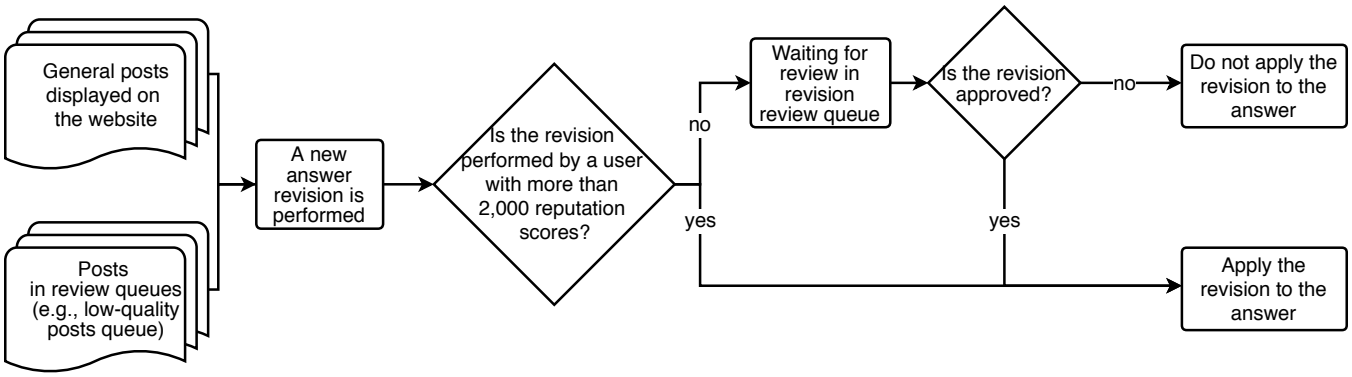

Fig. 2: The process of answer revision review.

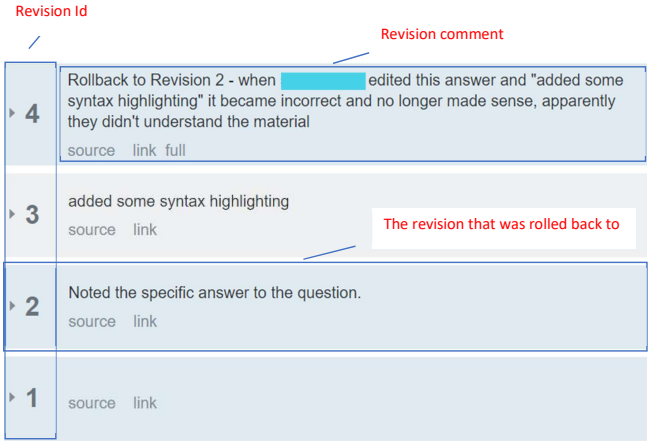

Fig. 4: An example of a rollback. The user explained their rationale for performing the rollback in the revision comment.

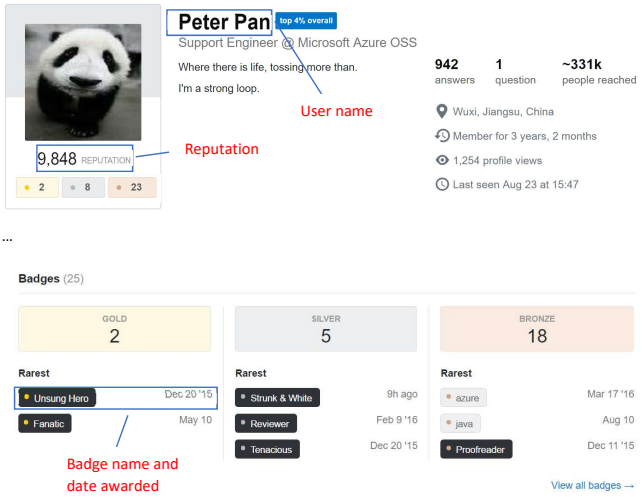

Fig. 5: An example of badges as shown in a user's online profile. The user was awarded 2 gold badges, 5 silver badges, and 18 bronze badges.

2.3 Motivating Answer Revisions Using a Badge System

Badge systems are widely used in various online systems, such as learning systems and Q&A websites [7], [8]. A badge is used as an indicator of accomplishment, skill, quality, or interest. For example, Quora employs several badges (e.g., Most Viewed Writer) to motivate users to write good quality posts. All websites (including technical and non-technical Q&A websites) under Stack Exchange[7] use the same badges system. In this study, we focus on the badge system of Stack Overflow since it is a website that is widely used by developers worldwide on daily basis. Stack Overflow uses a badge system to motivate users to participate in the community. Users can receive badges after completing specific goals (e.g., revising answers). Such badges are indicators of accomplishments, skills, or interests of a user. Badges have three different colors: gold, silver, and bronze, which indicate the level of difficulty to receive a badge. As an example, Figure 5 shows the badges that are listed in a user's profile.

We are interested in studying the badges that are related to revising answers. Table 1 lists the details of these badges: Strunk & White (silver), Archaeologist (silver), and Copy Editor (gold). Note that the badge "Copy Editor" excludes self-revisions, deleted posts, and tag edits, while both "Strunk & White" and "Archaeologist" take self-revisions into consideration. Thus, we study both self-revisions and revisions made by others in RQ1. We then focus on studying revisions by others in RQ2 and RQ3. We define the users who were awarded revision-related badges as *badge-holders* and the users who have never been awarded any revision-related badges as *non-badge-holders*. The current badge system encourages users to revise answers based on the quantity of revisions (e.g., by revising more than 500 answers). We are interested in examining how the badge system steers the answer revision process on Stack Overflow.

7. https://archive.org/details/stackexchange

3 Research Questions & Data Collection

In this section, we describe our research questions and motivation, as well as how we collect the datasets that we used to answer our research questions.

3.1 Research Questions

RQ1: Do badges change the number of user revisions over time as they are about to receive a badge?

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

TABLE 1: The badges that are awarded to badge-holders.

Badge	Description
Strunk & White (silver)	Revise 80 posts
Archaeologist (silver)	Revise 100 posts that were inactive for 6 months
Copy Editor (golden)	Revise 500 posts (excluding own or deleted posts and tag edits)

Note: In contract to the official descriptions of the badges, we change the word "edit" to "revise" in the badge descriptions to make our wording consistent throughout the paper.

The quality of user-generated content varies drastically on Stack Overflow. Some content contains valuable information while other content may contain undesired spam. A significant challenge for Stack Overflow is ensuring the quality of its content [2]. Hence, Stack Overflow encourages users to revise answers through a badge system (see Section 2.3). In this RQ, we investigate how badges steer the revision process. We study the revision patterns of users (e.g., do users revise answers in bursts or infrequently over a long period of time), especially when they receive revision-related badges. Do users still make revisions after getting such badges? We are also interested in investigating whether the users with revision-related badges exhibit different revision patterns compared with the users with no revision-related badges.

RQ2: How do users revise answers in terms of their content and size?

In this RQ, we investigate which parts (i.e., text block or code block) of the answers that users revise, as well as the underlying reasons for revisions. For example, is a revision performed to fix a bug in the code block or to refine a description in the text block? Which parts of the answers do users revise when they perform revisions in burst? Our empirical investigation would provide us with a deeper understanding of the kinds of revisions (in terms of content) that users perform, and offer insights about the efficacy of the revision badges.

RQ3: Which factors are associated with answer rollbacks?

We found that most of the revisions are not required to be reviewed for Stack Overflow policy (see Section 2). In addition, we found that some revisions were rolled back and that these rolled back revisions may be related to incorrect changes (see an example in Figure 4). It is important to reduce or avoid rolled back revisions. Hence, in this RQ, we investigate the factors that are associated with rollbacks. More specifically, we study the relationship between rollbacks and the number of revisions that are made by a user in a single day (i.e., RevisionsPerDay). We also examine the reasons behind the rollbacks. Our study of the rollbacks may help Stack Overflow designers and researchers propose solutions to improve the current badge system and potentially reduce rollbacks.

3.2 Getting Answer Revisions Data From Stack Overflow

To study answer revisions on Stack Overflow and answer our abovementioned RQs, we downloaded the data dump of Stack Overflow from the link that is provided by Stack Exchange[8], which is a network of Q&A websites on topics across varied fields (e.g., programming and education). The data dump contains detailed information about the posts (i.e., questions and answers). The data dump stores all the event history of each post (e.g., body edit, post deleted, and post closed), including the date of each event, the user who triggered the event, the comments on each event, and the changed post after each event.

For our study, we used all data posted before March 2017. There were a total of 13 million questions and 21 million answers in the downloaded data. Figure 6 presents the number of revisions made to the answers on Stack Overflow. From Figure 6, we note that 27.8% (5,897,479) of the answers on Stack Overflow have at least one revision, and 0.8% (167,823) of the answers have at least 5 revisions.

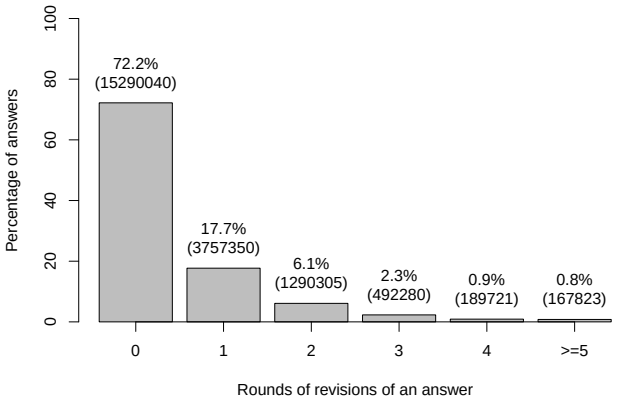

Fig. 6: The percentage of answers with a specific round of revisions on Stack Overflow.

3.3 Data Preprocessing

There are 38 types of events that are tracked by Stack Overflow[9]. In this study, we are interested in the "Edit Body" and "Rollback Body" events that are related to revisions that are performed on the body of an answer. An "Edit Body" event indicates that the body of an answer has changed. A "Rollback Body" event indicates that an answer's body has been reverted to a previous version. We consider the "Edit Body" and "Rollback Body" events that are performed on an answer as an answer revision (or revision for short) and as answer rollback (or a rollback for short) in this paper, respectively.

We select our studied answers based on the following criteria: 1) answers that are at least one year old; 2) answers with a score that is larger than 0. We choose such criteria to ensure that the studied answers have attracted enough attention from the community. We ended up with 2,377,692 answers and 3,871,966 corresponding revisions (17,156 Rollback Body and 3,854,810 Edit Body). 280,617 unique users were involved in these revisions.

8. https://archive.org/details/stackexchange

9. http://meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

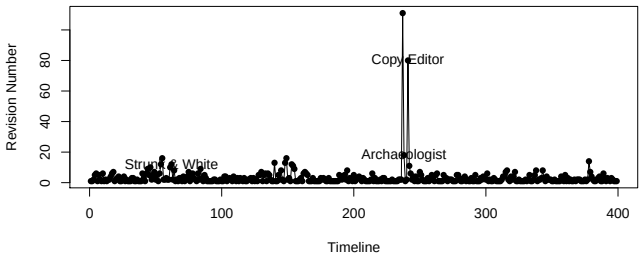

Fig. 7: An example of the revisions of a user over time. The badge-awarding days are marked in the plots. The user performed spikes of revisions on these badge-awarding days. The number of revisions that were performed on the badge-rewarding days are 16, 18, and 80 for Strunk & White, Archaeologist and Copy Editor, respectively. The threshold for spike is 14 days.

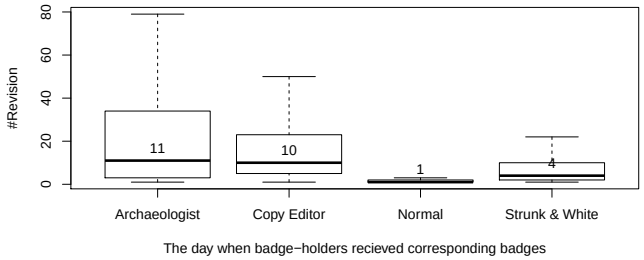

Fig. 8: The number of revisions that were performed by badge-holders on the days when they were awarded a badge (i.e., Archaeologist, Copy Editor, Strunk & White) versus normal days (i.e., days where they were not awarded a badge). The median values are marked in the figure.

4 Case Study Results

In this section, we present the results of our research questions. We discuss each research question along three parts: used approach, experimental findings, and a detailed discussion of our findings.

RQ1: Do Badges Change the Number of User Revisions over Time as They Are About to Receive a Badge?

Approach: To understand how badges steer the revision process of users, we investigate the revision patterns of users once they were awarded revision-related badges (i.e., badge-holders) over time. More specially, we look at how the number of revisions changes as users are about to receive revision-related badges and after obtaining a badge.

We compare the number of performed revisions on the day when a user received a badge (i.e., badge-awarding day) and a normal day (i.e., no badge is awarded). Note that we only consider the days when a user performed at least one revision. We perform a Mann-Whitney U test and a Cliff's d test [9] to determine whether or not the differences between the number of revisions between normal days and badge-awarding days are statistically significant and the magnitude of the differences. The effect size is assessed using the provided thresholds by Cliff [9]: $|d| < 0.147$ indicates that the effect size is negligible, $|d| < 0.33$ indicates that the effect size is small, $|d| < 0.474$ indicates that the effect size is medium, otherwise the effect size is large. To study the revision patterns of badge holders, we use the three-sigma rule (mean $\pm$ 3 $*$ standard deviation) [10], which is widely used for outlier detection [11], to identify the number of revision spikes over all users.

Results: ***Badge-holders performed considerably more revisions (statistically significant) on the badge-awarding days compared to normal days.*** Figure 8 shows a boxplot of the number of revisions that were performed by badge-holders on normal days and on badge-awarding days. We observed that the number of revisions that were performed on normal days (i.e., labeled as Normal on the figure with a median value of 1) is much less than those that were performed on badge-awarding days (i.e., Archaeologist, Copy Editor, and Strunk & White, and the median values are 11, 10, and 4, respectively). These observed spikes are compatible with previously documented phenomena in social psychology: people often escalate their efforts when they know that they are near their goal [12]. The Mann-Whitney U and Cliff's d results show that the differences between the two types are significant (p-value < 0.05) and large ($|d| > 0.474$). Moreover, we find that badge-holders performed spikes of revisions on 24.7% (i.e., 3,150) of their badge-awarding days. As an example, Figure 7 presents the revisions of a user over time[10]. We notice that there is always a spike of revisions around the badge-awarding days.

25% of badge-holders did not make any revisions after obtaining their one badge. Furthermore, we examine the revision activities of badge-holders once they received their first badge. We find that 77% of the badge-holders only have one revision-related badge. Among these badge-holders that only have one badge, 33% of them did not make any revision after obtaining one badge. Such phenomenon reflects that some users may be motivated by badges to perform answer revisions. Once they obtained a badge, they stopped performing revisions.

Badge-holders were 17 times more likely to perform spikes of revisions (i.e., perform a larger than usual number of revisions on a certain day) than non-badge-holders. To further understand the revision patterns of badge-holders, we compared the revision patterns with that of users who were never awarded a badge (i.e., non-badge-holders). We examine the number of revisions that were performed each day by both non-badge-holders and badge-holders. When considering the days in which users performed at least one revision, we observed that on average, badge-holders and non-badge-holders performed 3.0 (with a variance of 27.0) and 1.6 (with a variance of 1.9) revisions per day, respectively. To further examine such a high variance of 27.0, we study whether badge-holders are more likely to perform a large number of revisions on certain days (the threshold is 14 when using the three-sigma rule).

The revision spike detection results show that badge-holders have revision spikes on 2.7% (i.e., 73,325) of the days

10. stackoverflow.com/users/256793

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

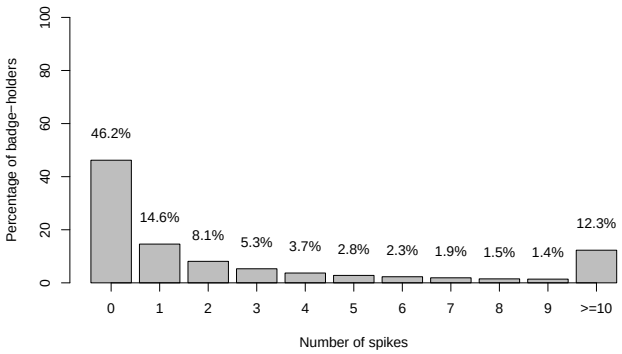

Fig. 9: The percentage of badge-holders that made x spikes of revisions in the studied Stack Overflow data (from the beginning to March 2017). 46.2% of the badge-holders never made spikes.

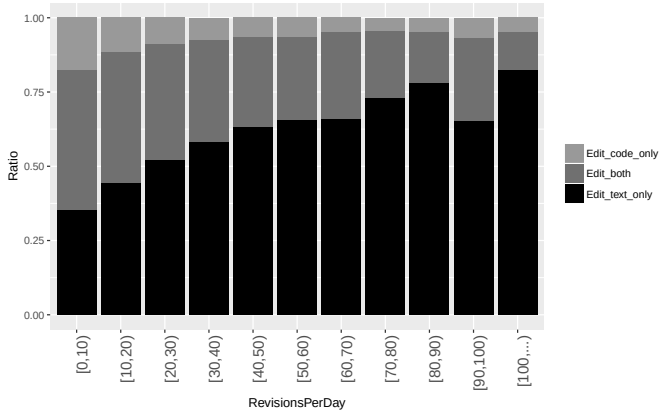

Fig. 10: The ratio of each type of revision against RevisionsPerDay. The ratio of Edit_Text_Only increases from 35.5% to 82.5% when the range of RevisionsPerDay increases from $0-10$ to ≥ 100. Note that the size of each bin is 3,538,679, 194,406, 37,552, 15,625, 8,309, 5,785, 3,356, 2,877, 2,019, 2,185, and 17,319. Same bin sizes for Figure 11.

when they performed revisions, while non-badge-holders have revision spikes on 0.16% (i.e., 6,751) of the days on which they performed revisions. In other words, badge-holders were 17 times more likely to have revision spikes than non-badge-holders. To further understand whether spikes are a rare occurrence for badge-holders, we draw a plot to show the number of spikes that were made by badge-holders against the percentage of such badge-holders (see Figure 9). We observed that 53.8% of the badge-holders made revision spikes, while only 1.3% of the non-badge-holders made revision spikes. 12.3% of the badge-holders made at least 10 spikes in total, which suggests that revision spikes are not a rare occurrence for badge-holders.

In summary, badge-holders performed a larger than usual number of revisions as they are about to receive a badge. Some users also stop revising posts after they receive their first revision-related badge. Our finding echoes with a prior study which finds that badges steer user behaviors (e.g., increasing participation) on Stack Overflow [3]. However, it is not clear what do users revise during such high intensity revision activities and how such high intensity activities affect the quality of revisions. Hence, in the next RQ, we study what users change when revising answers.

RQ2: How do Users Revise Answers in Terms of Their Content and Size?

Approach: We first conduct a quantitative analysis which examines whether the size and type of revisions change when the number of revisions that are performed in a single day increases. We then conduct a qualitative analysis to understand the underlying reasons for such revisions. Below, we describe the approaches that we use for our quantitative and qualitative analysis.

Quantitative Analysis

We are particularly interested in understanding whether there exists a relationship between the number of revisions that are performed in a single day (referred to as *RevisionsPerDay*) and the types (i.e., $Edit\_code\_only$, $Edit\_text\_only$, and $Edit\_both$) and the size of the revisions. We count the size of a revision by summing up the added and deleted characters in the revision. To study such relationships, we examine the ratio of the three types and the size of revisions against RevisionsPerDay and visualize our results. See more detailed description in Section A of Appendix.

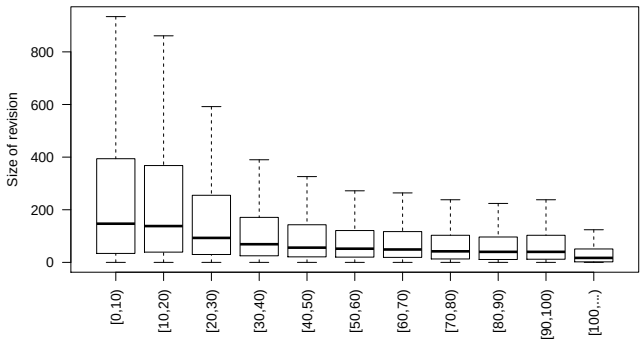

Fig. 11: The size of revisions against RevisionsPerDay. The median size of revisions (in characters) reduces from 147 to 17 when the range of RevisionsPerDay increases from $0-10$ to ≥ 100.

Results: ***Users were more likely to perform small or text revisions when they performed many revisions in a single day.*** We find that text revisions are prominent when users perform a large number of revisions in a single day. Figure 10 presents the ratio of the revision types against the number of revisions that were performed by a user in a single day. The general tendency is that the ratio of $Edit\_text\_only$ increases as the number of revisions that are performed in a single day increases. In other words, users were likely to revise more text than code when they performed many revisions in a single day.

Figure 11 presents the size of revisions against the num-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

ber of revisions that were performed by a user in a single day. The general tendency is that the size of revisions drops as RevisionsPerDay increases, which implies that users were likely to perform small revisions when they performed many revisions in a single day. One possible reason behind this is that users prefer to perform simple revisions so that they are able to perform as many revisions as possible in a single day. See additional results of the quantitative analysis in Section A in the Appendix.

Qualitative Analysis

We conducted a qualitative study to uncover the reasons behind answer revisions. In order to achieve a confidence level of 95% with a confidence interval of 5% [13], we randomly sampled 384 revisions from the entire revision data set (i.e., 3,871,966 revisions) and identified the rationale for such revisions. To compute the size of our random sample, we use the following formula $\frac{Nz^2p(1-p)}{e^2N+z^2p(1-p)}$, where N is the population size (e.g., 3,871,966), z is the Z-score corresponding to a particular confidence level (e.g., 1.96 for a confidence level of 95%), e is the confidence interval (e.g., 5%), and p is population proportion (e.g., 0.5). We first examined the randomly sampled revisions with no particular types of reasons in mind. Then, we qualitatively analyzed the sampled data and derived a set of reasons for these revisions (e.g., coding rules). Then, the first two authors manually checked the sampled revisions and categorized the sampled revisions based on the derived reasons. We used Cohen's kappa [14] to measure the inter-rater agreement. Our kappa value is 0.89, which implies a high level of agreement. Any discrepancies were discussed until a consensus was reached. During the qualitative analysis, the authors also needed to read comments that are posted under answers, which helped the authors identify the reasons more accurately. We calculated some basic statistics of the number of revisions for each reason and visualize the results.

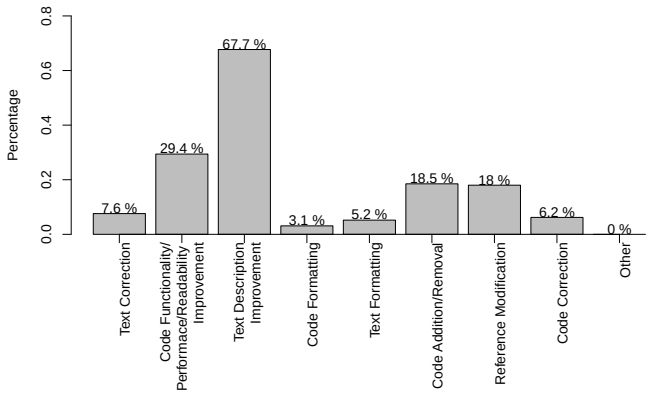

Fig. 12: The distribution of the reasons behind a revision based on the randomly sampled revisions. 66.7% of the revisions were performed to improve the text description of answers.

Results: ***Users were more likely to perform Text Correction and Code Formatting on other users' answers; it was very rare for a user to help others with Code Correction.*** Table 2 shows the eight types of reasons that we derived based on the sampled data. We labeled each sampled revision using these types and visualized the distribution in Figure 12. Note that one revision might have multiple types, since a revision probably changes multiple places and these places might be changed for different reasons. The percentage of a particular type (t) is calculated as $\frac{\#\ of\ revisions\ that\ are\ labeled\ with\ t}{\#\ of\ all\ revisions}$. We observed that the most common type of a revision is Text Improvement (67.7%), which indicates that most revisions are made to make the answer more concise and clear.

TABLE 2: An overview of the manually derived reasons (coding rules).

Revision Reason	Definition
Text Correction	Fix errors in text blocks. Such as fixing grammar/typo issue.
Text Description Improvement	Improve text description to make the description more clear and concise.
Text Formatting	Format the text to make a better presentation. Such as changing font, highlighting text, and removing/adding space.
Code Correction	Fix errors in code blocks. Such as fixing a bug and/or fixing a typo.
Code Functionality/Performance/ Readability Improvement	Improve code in terms of functionality, performance, readability. Such as changing function, changing logic, adding comment, changing type, and changing variable name.
Code Formatting	Improve code in terms of formatting. Such as adding/removing space, and adding newline.
Code Addition/Removal	Add/remove code snippets. Such as adding/removing entire block of code or adding/removing line(s) of code from code block.
Reference Modification	Add/update/remove references from text block, like url, hyperlink, and image.

Figure 13 shows the percentages of revisions that were made by the answer creator (creator) versus the revisions that were made by other users (helpers) for each reason type. We observed that helpers were more likely to help with Text Correction and Code Formatting. On the other hand, almost all Code Correction revisions were made by the answer creators themselves. It was also rare for users (i.e., non-creators) to help others improve code.

One possible reason for our abovementioned observation is that correcting the code requires deep knowledge of the question/answer. Thus, it may be harder for helpers to make Code Correction revisions. A similar phenomenon was also observed in collaborative code review task, where reviewers tend to provide shallow feedback [15]. Based on this observation, we may suggest that Stack Overflow consider encouraging users to perform code-related revisions since previous studies have shown that code snippets are an important aspect of high-quality answers [16], [17], [18].

RQ3: Which Factors are Associated with Answer Rollbacks?

Approach: We first conduct a quantitative analysis to understand the relation between the number of revisions per day and rollbacks. We then conduct a qualitative analysis to understand the underlying reasons for such rollbacks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

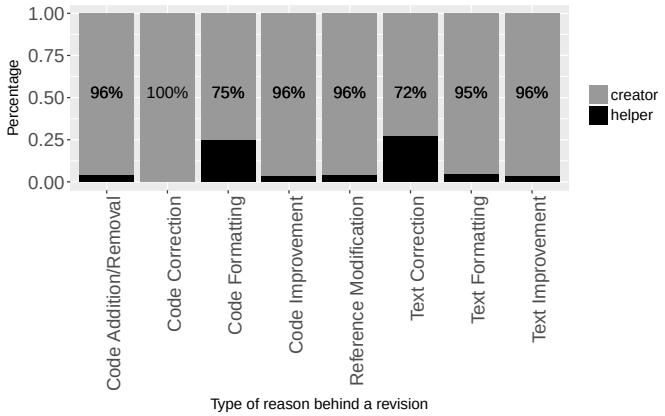

Fig. 13: Percentages of revisions that are made by the answer creator (creator) v.s. revisions that are made by other users (helper) in each type of reason behind a revision based on our randomly sampled and manually labelled revisions.

Below, we describe the approaches that we use for our quantitative and qualitative analysis.

We constructed our studied dataset using the following two steps:

1) **Identifying the revisions that were rolled back.** We consider all the revisions that happened between the rolled back revision and the revision to which it is rolled back as the revisions that result in rollbacks (referred as *unaccepted revision*). For example, in Figure 4, the rollback reverts the answer from revision 4 to revision 2, then revision 3, which is between revision 2 and 4, is considered as an unaccepted revision.
2) **Removing self-reverted revisions.** There are two scenarios of a rollback: 1) users are not satisfied with *their own revision* and thus they rolls back the answer to a previous version; 2) users are not satisfied with *other users' revisions* and the answer creator rollback the answer to a previous version. We focus our study on the second scenario.

After our data preprocessing steps, we ended up with 9,087 unaccepted revisions and their corresponding rolled back revisions.

Similar to the approach that we used in RQ2, we first conducted a quantitative analysis to understand the relationship between rollbacks and the number of revisions that are performed in a single day (i.e., RevisionsPerDay). We then conducted a qualitative analysis to understand the underlying reasons for the rollbacks. We describe below the approaches that we use for our quantitative and qualitative analysis, as well as our results.

Quantitative Analysis

To understand the relationship between the rollbacks and RevisionsPerDay, we compared the RevisionsPerDay for the days when a user performs unaccepted (i.e., rolled back) revisions (referred as $RevisionsPerDay_{unaccepted}$) and the RevisionsPerDay of the days when the *same user* performs *no* unaccepted revisions (referred as $RevisionsPerDay_{accepted}$). If $RevisionsPerDay_{unaccepted}$ is significantly larger than $RevisionsPerDay_{accepted}$, it may indicate that making more revisions a day will more likely result in rollbacks.

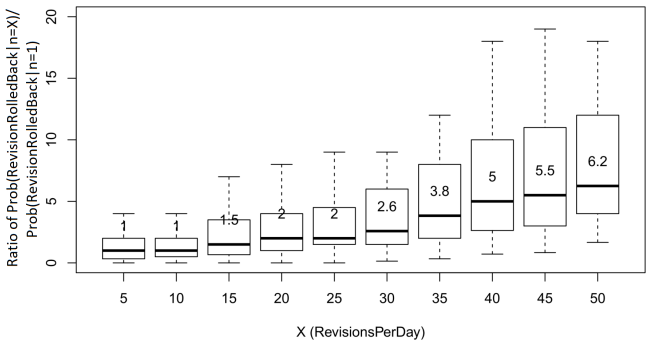

Fig. 14: Ratios of the probability of a revision being rolled back on the day when RevisionsPerDay is larger than X ($Prob(RevisionRolledBack \mid n > X)$ and that of when RevisionsPerDay is 1 ($Prob(RevisionRolledBack \mid n = 1)$), which is 0.2%. The likelihood of a revision being rolled back increased as more revisions were performed in a single day. 1,628,152, 528,607, 56,703, 29,885, 19,332, 14,124, 11,357, 9,511, 8,106, and 6,803 revisions on the day when X is 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50, respectively.

We calculated the probability of a revision being rolled back given different values for RevisionsPerDay (based on all users). The probability (i.e., $Prob(RevisionRolledBack \mid n = N)$) of a revision being rolled back when a user performs N revisions on a particular day (i.e., $RevisionsPerDay_N$) could be estimated in the following way using bootstrap [19]. For each iteration i: 1) We randomly sample s (i.e., $s = 1000$ in this study) revisions (with replacement) from all revisions that were performed on the $RevisionsPerDay_N$; 2) We calculate the probability (i.e., $Prob_i(N)$) of having unaccepted revisions among these s revisions that are sampled in step 1. We repeat the iteration 100 times. We use these values $\{Prob_1(N),...,Prob_i(N),...,Prob_{100}(N)\}$ to estimate $Prob(RevisionRolledBack \mid n = N)$. To further understand the relationship between RevisionsPerDay and the probability of a revision being rolled back, we calculate the ratios of $Prob(RevisionRolledback \mid n > X)$ and $Prob(RevisionRolledBack \mid n = 1)$, where X is a different value for RevisionsPerDay (i.e., 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50). We present the ratios in a figure. We also performed Mann-Whitney U test to measure whether the difference between $Prob(RevisionRolledback|n = 1)$ and $Prob(RevisionRolledback|n > x)$ are statistically significant. We performed a Cliff's d test to measure the magnitude of the differences.

Results: ***Making more revisions in a single day increased the likelihood of a revision being rolled back (e.g., due to lower revision quality), especially when the number of revisions is large (e.g., larger than 50).*** The mean value of $RevisionsPerDay_{unaccepted}$ (i.e., 3.6) is almost 1.4 times larger than that of $RevisionsPerDay_{accepted}$ (i.e., 2.5). We performed a Mann-Whitney U test and compute Cliff's d. We observed that the differences between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

$RevisionsPerDay_{accepted}$ and $RevisionsPerDay_{unaccepted}$ are statistically significant ($p-$value < 0.5) with a large effect size (Cliff's d > 0.474), which implies that unaccepted revisions are usually performed on days when a user performs more revisions than usual. In other words, making more revisions in a single day increases the likelihood of a rollback.

Figure 14 further presents the ratios of the probability of a revision being rolled back on the days when the number of RevisionsPerDay is larger than X ($Prob(RevisionRolledBack \mid n > X)$, where $X = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$) and that of when the number of RevisionsPerDay is 1 ($Prob(RevisionRolledBack \mid n = 1)$). We observed that the likelihood of a revision being rolled back increases as the number of RevisionsPerDay increases. When the number of RevisionsPerDay is larger than 50, the likelihood of rollbacks (i.e., 1.2%) is 6 times larger than that of when the number of RevisionsPerDay is 1 (i.e., 0.2%). In addition, the results of Mann-Whitney U test and Cliff's d show that $Prob(RevisionRolledBack \mid n > X)$ is statistically higher than $Prob(RevisionRolledBack \mid n = 1)$ ($p-$value < 0.5) with a large effect size (Cliff's d > 0.474) when X is larger than 10.

One possible reason is that when a user performs considerably more revisions than usual in a single day, the user may have difficulties ensuring the quality of every revision (e.g., due to their limited time), hence increasing the likelihood of low-quality revisions that are eventually rolled back.

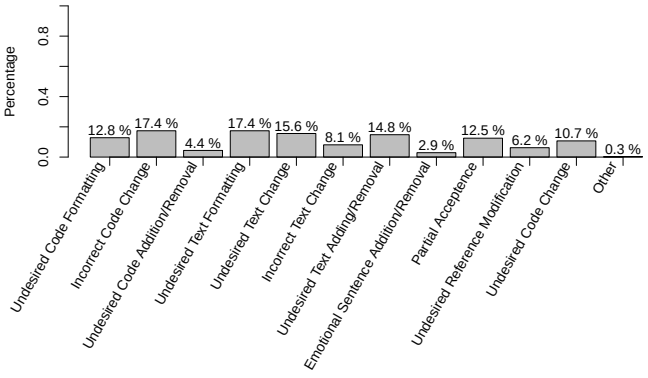

Fig. 15: An overview of the types in which the qualitatively-studied rollbacks occur. For example, among the studied unaccepted revisions, 17.4% of the revisions were rejected due to Incorrect Code Change.

Qualitative Analysis

To further understand the reason behind rollbacks, we investigated the relationship between the number revisions that are performed in a single day and rollbacks. We conducted a qualitative analysis by randomly sampling 369 cases from the 9,087 rollbacks using a 95% confidence level with a 5% confidence interval. We performed a lightweight open coding-like process [20], [21] for identifying the reasons behind rollbacks (see Section B.1 for details).

TABLE 3: An overview of the manually derived reasons for rolling back a revision.

Rollback Reasons	Definition
Undesired Code Formatting	Users made undesired code formatting changes, such as adding/removing space, and adding a newline.
Incorrect Code Change	Users made incorrect code changes.
Undesired Code Addition/Removal	Users added/removed undesired code, e.g., adding alternative solutions, and removing a code snippet.
Undesired Code Change	Users made undesired code changes, such as changing options of a command, changing APIs, refactoring, and editing comments.
Undesired Text Formatting	Users made undesired text format changes, such as changing font, highlighting text, and removing/adding space.
Incorrect Text Change	Users made incorrect text changes, such as alternating the meaning of the sentence.
Undesired Text Addition/Removal	Users added/removed undesired text, such as adding additional solutions and adding advertisement of a tool.
Undesired Text Change	Users made undesired text changes such as changing the structure of a paragraph and rewording.
Partial Acceptance	The revision is rolled back, but part of the changes are still accepted and are included in a later revision.
Emotional Sentence Addition/Removal	Users added/removed sentences that present their personal emotion.
Undesired Reference Modification	Users modified/added/removed the references (e.g., links and images) in the answer.
Other	Other modifications, e.g., an asker asked a question in the answer.

Results: ***Answer revisions were often rolled back due to undesired code/text formatting, incorrect code revisions, and other text-related revisions.*** Figure 15 presents an overview of the reasons that users rollback answer revisions. We observed that 30.2% of the answers were rolled back because of undesired formatting (i.e., 12.8% Undesired Code Formatting and 17.4% Undesired Text Formatting). Based on our qualitative analysis, one possible reason that users rolled back formatting revisions is that users do not like revisors to change the formatting because such revisions may make the answer look ugly (from the perspective of the answer creator) or may even change the intention that the answer creator wished to express (e.g., emphasize or deemphasize parts of an answer).

Figure 16 presents an example of Undesired Text Formatting[11]. The user mentioned in the revision comment why he did not like the edits. The user thought that "gVim" looks totally ugly and did not like the formatting change of "internet". The user also did not like the formatting of the command mode "g" and "t".

We observed that most revisions (more than 38.5%) were rolled back due to text-related revisions (i.e., Undesired Text Change, Incorrect Text Change, and Undesired Text Addition/Removal). Helpers may misunderstand the answer and thus make incorrect revisions. In addition, some revisions may make the answer deviate from the original purpose. However, when manually checking the sampled

11. http://stackoverflow.com/posts/24156/revisions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

rollbacks, we did not see any rollbacks due to fixing grammar/typo issues. We also observed that many revisions (17.4%) were rolled back due to Incorrect Code Change. For example, an answer creator rolled back the previous revision and mentioned "*I am reverting back to my original question text. The latest edit actually renders my code* ***incorrect****, as '<?php Form::...' doesn't echo anything to the browser*"[12].

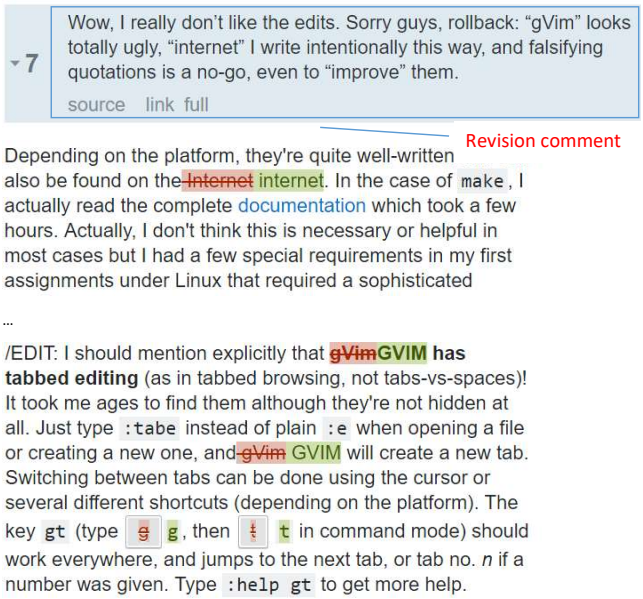

Fig. 16: An example of an Undesired Text Formatting.

See additional results in Section B in the appendix.

5 Discussion

In this section, we discuss the feedback that we collected from the Stack Overflow Meta forum[13]. We also discuss the implications of our findings.

5.1 Feedback From the Stack Overflow Meta Forum

To understand whether our research uncovered a relevant problem on Stack Overflow and search for possible suggestions to resolve or alleviate the problem, we posted our findings on the Stack Overflow Meta forum, which is a Stack Overflow forum where the Stack Overflow community members (i.e., developers and users of Stack Overflow) discuss the inner-workings and policies of Stack Overflow. Stack Overflow encourages members of its community to leave feedback on its Meta forum so that Stack Overflow can improve its website based on feedback (e.g., feature request).

As of press time, our post[14] was ranked as the top 7% questions among all badge-related questions. Our post received 45 upvotes, 3 favorite votes, and 666 views. The median upvotes, favorite votes, and views on Stack Overflow Meta forum are 4, 1, 146, respectively.

12. https://stackoverflow.com/posts/14756994/revisions
13. https://stackoverflow.com/help/whats-meta
14. https://meta.stackoverflow.com/questions/350326/should-the-post-revision-related-badges-e-g-copy-editor-and-strunk-white-b/350426#350426

Our post received useful feedback from the Stack Overflow community. The community members felt that our study touched on a real challenge for Stack Overflow. One community member left a comment "SO has rate-limits on almost anything you do, specifically to avoid ab/use like this. But not on edits for some rather mysterious reason. Plenty of other trouble caused by this, from DoS attacks on users' Interesting page view to very current problems with the edit review queue". The poster felt that no rate-limitation on revisions could even raise some security issue. Another community member felt our findings were interesting (i.e., "Ah well, it's a lot of good research already.") and he also asked whether this is a concern for other types of badges (e.g., task review badges).

Many useful suggestions were provided by the community members and these suggestions could be categorized into two major categories. First, community members proposed the use of a rating (voting) mechanism on revisions and consider the rating of a revision as part of a badge. For instance, one community member mentioned "I've always wanted to be able to upvote good edits. Perhaps something along those lines could be used to help determine high-quality edits. Coupled with number of edits, it would be similar to a tag badge (e.g. 20 edits with a total score of 20)". Another community member mentioned "As far as quantity vs quality, it seems to me that the only moderately simple way to quantify quality is to put in a voting mechanism for edits themselves. Then allow only $>$ 2000 rep users to vote on that. Of course something like that would be complicated and likely have its own unintended consequences".

Second, community members suggested to not consider rolled back revisions when awarding badges. For example, "Considering the point about rollbacks, one way to discourage the "quantity over quality" behavior would be to have the badges not credit any edit submitted on the same day as a rolled back edit (or alternatively, no credit for edits submitted within 24 hours following submission of an edit later rolled back, since this can be computed using a single pass through the edit history).".

Another interesting suggestion is to add a rollback-message feature. One community member mentioned that Stack Overflow does not notify users if their revisions are rolled back, i.e., "your post is focusing on rollbacks, but as a revisionist (12,433 posts edited), I have no way to know which of my edits got rollbacked, so I can't improve myself on that. I believe we should first improve on communicating on when an edit got rollbacked. And possibly why an edit got rollbacked by eventually adding a rollback-message feature."

In short, our findings reveal challenges (e.g., no limitation on the number of revisions that are made per day when awarding badges, no quality control for revisions) in the mechanisms of Stack Overflow and attracted the attention from the Stack Overflow community. Future research should explore ways to improve revision-related badges.

5.2 Implication and Highlights of Our Findings

We observed that some users tended to have a spike of revisions just right before getting badges. This observation is compatible with the finding by Ashton et al. [3], in which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

they found that the activities of users increase substantially before users achieve a badge. However, the current revision related badges only consider the quantity of revisions and fail to consider the quality of revisions. The Stack Overflow website designers may consider improving their badge system to ensure the revision quality. Below, we elaborate our findings and the corresponding implications for Stack Overflow website designers.

Certain rules may prevent users from gaming the badge system and provide a better balance between the quality and quantity of revisions that are performed by users. We find that making more revisions a day increased the likelihood of rollbacks, especially when making considerably more (e.g., over 50) revisions than usual. To reduce the chances of rollbacks, we suggest that Stack Overflow to consider developing certain rules to prevent users from making considerably more revisions than usual in a single day (e.g., gaming the badge system by performing over 100 revisions per day). For example, Stack Overflow website designers may set up a maximum number of daily revisions that could be counted toward receiving a badge. Alternatively, Stack Overflow might choose not to limit revisions and instead only count unrolled revisions. Another suggestion is to employ a rating mechanism to improve the quality of revisions. Note that any rule may have undesirable side effects. For example, punishing rollbacks or voting the quality of revisions may affect the user participation in revising activities. Hence, future studies should investigate ways to ensure a good balance between both the quality and quantity of revisions while minimizing the impact on user participation.

Stack Overflow designers should consider encouraging users to perform code-related revisions by designing new badges or making the current badges favour code revisions. We observed that users tended to perform text revision rather than code revision when they made relatively more revisions in a single day. In addition, users were more likely to perform Text Correction and Code Formatting on other users' answers; it was rare for users to help others correct or improve code snippets in answers. However, previous studies have shown that code snippets are an important aspect of high-quality answers [16], [17], [18]. Thus, we suggest that Stack Overflow website designers may consider encouraging users to perform code-related revisions. For example, Stack Overflow website designers could design new badges that are related to code revisions or modify the current badges to favour code revisions (e.g., making a correct code revision equivalent to several simple text revisions or trivial code formatting revisions).

6 Threats to Validity

External validity. Threats to external validity relate to the generalizability of our findings. In this study, we focus on Stack Overflow, which is one of the most popular Q&A websites for developers, hence, our results may not generalize to other Q&A websites (e.g., non-technical Q&A websites under Stack Exchange that do not focus on software development), such as Code Project [15] and Photography[16]. To alleviate this threat, more Q&A websites could be studied in the future.

We conducted several qualitative analysis in our RQs; however, it is impossible to manually study all revisions. To minimize the bias when conducting our qualitative analysis, we took statistically representative samples of all relevant revisions with a 95% confidence level and a 5% confidence interval [13] as what was done in prior studies [22], [23] (we ended up studying 384 randomly sampled revisions in RQ2 and 369 randomly sampled unaccepted revisions in RQ3).

Internal validity. Threats to internal validity relate to experimenter bias and errors. Our study involved qualitative analysis of revisions in RQ2 and RQ3. To reduce the bias, each revision was labeled by two of the authors and discrepancies were discussed until a consensus was reached. We also showed that the level of inter-rate agreement of the qualitative studies is high.

In this study, we detect source code in an answer by using existing HTML tags "$< pre >$ $< /pre >$ and "$< code >$ ... $< /code >$, which are recommended by Stack Overflow for users to format any code in their posts. There is not guarantee that all users will format all their code using the recommended HTML tags. This may cause a threat to the validity of our study.

Construct validity. One construct threat is that it is difficult to find data that could directly show the quality of revision. Hence, we use rollbacks, which we think are a reasonable and basic measure for capturing the quality of a revision.

7 Related Work

In this section, we discuss related work to our study. We focus on three related topics: collaboration on Q&A websites, understanding and improving question quality, understanding incentive systems.

7.1 Collaborative Editing

Collaborative editing has been used in many areas, such as online knowledge database editing [24], [25], science collaboration [31], [32], and software development [33], [34]. Zhu et al. examined the collaborative editing of posts (i.e., both answer and question) on Stack Overflow, and explored its benefits on content quality and potential negative effects on users activity [24]. They found that collaborative editing could improve the number of positive votes, which implies an increase of the quality of posts. Different from their findings that collaborative editing improves the quality of posts on Stack Overflow, our findings show that making considerably more revisions than usual in a single day may decrease the quality of answers (i.e., rollback). Munteanu et al. presented a design of a webcast extension that engages users to collaborate in a Wiki-like manner on editing the transcripts that are produced by automatic speech recognition techniques [35]. Munteanu et al. showed that this is a feasible solution to improve the quality of transcripts [35]. Kittur et al. examined how the number of editors on Wikipedia and the coordination methods that they used affect the quality of Wikipedia article [25]. They observed that adding more editors has no association with

15. https://www.codeproject.com/
16. https://photo.stackexchange.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

TABLE 4: Comparison between our findings and findings of prior studies.

	Our study	Prior study	Comparison
Collaborative Editing	Collaborative editing does not always lead to higher quality answers. Revisors who perform considerably more revisions than usual in a single day may negatively affect the quality of answers (i.e., cause rollbacks).	Zhu et al. found that collaborative editing can improve the quality of posts on Stack Overflow [24].	Our study focused on particular cases (e.g., spike revisions), while they measured the overall impact of collaborative editing.
		Kittur et al. found that adding more editors has no association with improvements in the quality of articles when the work was distributed evenly among editors or when they used explicit communication on the article talk page to coordinate on Wikipedia [25].	Our study echoes their findings to some extent: inappropriate collaborative editing may not improve the quality of answers.
Understanding and Improving the Quality on Q&A Websites	Making considerably more revisions than usual in a single day increased the likelihood of a revision being rolled back. Answer revisions are often rolled back due to undesired code/text formatting, incorrect code revisions, and other text-related revisions.	Prior studies examined the quality of articles on Q&A websites based on the text, code, and user information of the article itself [2], [17], [26], [27], [28].	Prior studies never made use of revision information to examine the quality of posts on Q&A websites. Our findings suggest that the revision information (e.g., the spike of revisions) can also be leveraged for such studies.
		Hu et al. made use of the interaction data between articles and their contributors as derived from the article edit history (e.g., review behavior, author authority) to measure the quality of articles on Wikipedia [29].	
Understanding Incentive Systems	Badge-holders were considerably likely to perform spikes of revisions than non-badge-holders, especially on the day when the users were awarded badges.	Prior studies confirmed the value of incentive systems and their effectiveness on user participation on Q&A websites [3], [4], [30].	Our finding is compatible with prior studies.

improvements in the quality of articles, especially when the work was distributed evenly among editors or when they used explicit communication on the article talk page to coordinate. Our study echoes with their observations to some extent: collaborative editing (e.g., revision) does not always improve the quality of an answer. Calvo et al. proposed an architecture for supporting collaborative editing for academic writing [32]. They analyzed the impact of writing activities on the quality of outcomes.

These prior studies mainly focused on investigating the impact of collaborative editing on the quality of user-generated contents (e.g., answer posts) and found that collaborative editing could improve the quality of articles in general. Our study focused on how users revise answers on Stack Overflow and found that in certain situations (i.e., a collaborator making considerably more revisions than usual in a single day), the edits that the collaborator made may reduce the quality of articles. Moreover, we measured the quality of revisions by looking at rollbacks rather than positive votes, which may be impacted by many confounders (e.g., the usefulness of answers). Table 4 highlights the comparison of our findings and the findings of prior studies.

7.2 Understanding and Improving the Quality on Q&A Websites

The quality of user-generated content varies drastically on crowdsourcing websites (e.g., Q&A websites). Some content may contain valuable information while other content may contain unwanted spam. One significant challenge that Q&A websites have is to ensure the quality of their content [2]. Therefore, numerous studies have been done to understand and improve the quality of posts on Q&A websites. Asaduzzaman et al. performed a study on the unanswered questions on Stack Overflow and found that some questions did not receive any answer due to the question being too short, not clear, too hard, or unrelated (not related to the Stack Overflow community) [26]. Rahman and Roy studied five aspects (i.e., answer rejection rate, last access delay, topic entropy, reputation and vote) that are associated with unresolved questions and built models based on these five aspects for understanding unresolved questions [36]. Hudson et al. investigated the characteristics of questions for which clarification are requested (e.g., missing information, unclear goals, non-standard terminology) from helpers on Q&A websites. Hudson et al. observed a large difference of such characteristics across different websites [37]. Ponzanelli et al. performed an empirical study to understand the relationship between a set of proposed factors and the quality of a post on Stack Overflow [17]. Ponzanelli et al. also built a classification model to identify high-quality and low-quality questions as soon as questions are posted [38]. Yao et al. found that the quality of an answer is highly associated with that of its question [39]. Thus, based on this observation, they proposed algorithms to predict the quality of questions and answers. Similarly, Yao et al. also proposed a family of algorithms to identify high-quality posts on Q&A websites based on the correlation between an answer and its question [40]. Harper et al. investigated predictors of answer quality on Yahoo! Answers [41]. They found that paying money for an answer led to better outcomes. Li et al. investigated the factors that may impact the quality of answers on ResearchGate and found that authority of responders, shorter response time and greater answer length are positively associated with the quality of answers [42]. Duijn et al. investigated the impact of various text-related and code-related factors on the quality of a question on Stack Overflow and found that the code to text ratio of a question is the most important factor [27]. Calefato et al. investigate the factors that impact the success of a question (i.e., receiving an accepted answer) and performed a survey to collect guideline suggestions for writing questions on Stack Overflow [28]. They provided

guidelines for writing questions on Stack Overflow, such as write questions using a neutral emotional style, and provide sample code and data.

Prior studies focus on exploring the factors that affect the quality of posts and on proposing approaches to estimate the quality of posts on Q&A websites. However prior studies did not consider the information derived from the answer revision when studying the quality of posts. Our findings may indicate that the revision information (e.g., whether a post has a revision that is made by a user on the day when he/she makes a spike of revisions) probably could be leveraged to measure the quality of posts on Q&A sites. Similar to a prior study by Hu et al. [29], which proposed three quality measurement models that make use of the interaction data between articles and their contributors as derived from the article edit history (e.g., review behavior, author authority) to measure the quality of articles on Wikipedia.

7.3 Understanding Incentive Systems

A number of studies focus on studying the incentive systems of Q&A website. Cavusoglu et al. performed an empirical study on the incentive system on Stack Overflow and provided evidence to confirm the value of the incentive system and its effectiveness on increasing user participation [30]. Anderson et al. studied how user behavior is steered by the badge system on Stack Overflow [3]. They observed that a badge can increase the overall level of user participation on the site and the extent of steering depends on how close the user is to the badge boundary. Grant et al. also observed that badges can be used to influence user behaviour by demonstrating an increase in user activity related to a badge immediately before it is awarded [4]. We obtained similar observations in our study (e.g., in RQ1, we observed that users were more likely to perform spikes of revisions right before getting badges). Antin and Churchill analyzed badge systems in social media from the psychological perspective and presented five social psychological functions of badge systems: goal setting, instruction, reputation, status/affirmation, and group identification [43]. Hsieh et al. investigated the impact of financial incentives on Q&A websites and they observed that paying more may elicit a larger number of answers but may not elicit higher quality answers [5]. Jan et al. examined how financial incentives affect different players in social Q&A services [6]. They found that financial incentives attract answers faster from experts, but the incentives also drive certain users to aggressively game the system for profits. Wang et al. explored how one may improve the current incentive systems to motivate fast answering of questions [44]. They suggested that Q&A sites should improve their incentive systems to motivate non-frequent answerers to be more active.

Our study is different from the above-mentioned studies, which mostly focused on understanding the incentive system. Our study focused on investigating how users revise answers on Stack Overflow and how badges may affect revision activities. We also provided some suggestions on how Stack Overflow designers may improve the badge systems according to our findings.

8 Conclusion

Stack Overflow employs an incentive system that motivates users (by awarding badges to users) to continuously improve and maintain the quality of answers. Such answer revision activities are very common on Stack Overflow. We found that more than 25% of the answers were revised after they were initially posted, which implies that answer revision is a major activity on Stack Overflow.

In this study, we analyzed 3,871,966 revisions that were collected from 2,377,692 answers to understand how the users revise answers and the impact of those revisions. We found that badge-holders performed considerably more revisions (statistically significant) on the badge-awarding days compared to normal days. We also found that revisions that were performed during such spikes are more likely to be rolled back. In addition, users were more likely to perform minor and non-code revisions, especially when they performed many revisions in a single day. Moreover, we shared our observations with the Stack Overflow community, who agreed with our observations and led to a discussion of proposing several ways to improve the current badge system.

In short, the current badge system on Stack Overflow is designed to ensure the quantity of revisions (i.e., badges are awarded according to quantitative measures such as the number of revisions), however such a badge system fails to consider the quality of revisions. Thus, Stack Overflow designers may consider to improve their badge system to provide a better balance between the quality and quantity of revisions. Stack Overflow designers may also consider encouraging users to revise code by designing new badges that are related to code revisions or changing current badges to make them favour code revisions.

References

[1] D. Ford, K. Lustig, J. Banks, and C. Parnin, "We don't do that here: How collaborative editing with mentors improves engagement in social Q&A communities," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, 2018, pp. 608:1–608:12.

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding high-quality content in social media," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ser. WSDM '08, 2008, pp. 183–194.

[3] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Steering user behavior with badges," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13, 2013, pp. 95–106.

[4] S. Grant and B. Betts, "Encouraging user behaviour with achievements: An empirical study," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR '13, 2013, pp. 65–68.

[5] G. Hsieh, R. E. Kraut, and S. E. Hudson, "Why pay?: Exploring how financial incentives are used for question & answer," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, 2010, pp. 305–314.

[6] S. T. Jan, C. Wang, Q. Zhang, and G. Wang, "Towards monetary incentives in social q&a services," *arXiv preprint arXiv:1703.01333*, 2017.

[7] R. Shields and R. Chugh, "Digital badges – rewards for learning?" *Education and Information Technologies*, vol. 22, no. 4, pp. 1817–1824, Jul 2017.

[8] J. Jones and N. Altadonna, "We don't need no stinkin' badges: examining the social role of badges in the huffington post." in *CSCW*, S. E. Poltrock, C. Simone, J. Grudin, G. Mark, and J. Riedl, Eds. ACM, 2012, pp. 249–252.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

[9] N. Cliff, "Dominance statistics: Ordinal analyses to answer ordinal questions." *Psychological Bulletin*, vol. 114, no. 3, pp. 494–509, Nov. 1993.

[10] D. Wheeler, D. Chambers, and D. Chambers, *Understanding Statistical Process Control*. Statistical Process Controls, 1992.

[11] T. H. D. Nguyen, B. Adams, Z. M. Jiang, A. E. Hassan, M. Nasser, and P. Flora, "Automated verification of load tests using control charts," in *Proceedings of the 2011 18th Asia-Pacific Software Engineering Conference*, ser. APSEC '11, 2011, pp. 282–289.

[12] S. Fox and M. Hoffman, "Escalation behavior as a specific case of goal-directed activity: A persistence paradigm," in *Journal Basic and Applied Social Psychology*, 2002, pp. 273–285.

[13] S. Boslaugh and P. Watters, *Statistics in a Nutshell: A Desktop Quick Reference*, ser. In a Nutshell (O'Reilly). O'Reilly Media, 2008. [Online]. Available: https://books.google.ca/books?id=ZnhgO65Pyl4C

[14] K. Gwet *et al.*, "Inter-rater reliability: dependency on trait prevalence and marginal homogeneity," *Statistical Methods for Inter-Rater Reliability Assessment Series*, vol. 2, pp. 1–9, 2002.

[15] A. Bacchelli and C. Bird, "Expectations, outcomes, and challenges of modern code review," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE '13, 2013, pp. 712–721.

[16] F. Calefato, F. Lanubile, M. C. Marasciulo, and N. Novielli, "Mining successful answers in stack overflow," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR '15, 2015, pp. 430–433.

[17] L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza, "Understanding and classifying the quality of technical forum questions," in *Proceedings of the 2014 14th International Conference on Quality Software*, ser. QSIC '14, 2014, pp. 343–352.

[18] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web? (nier track)," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11, 2011, pp. 804–807.

[19] B. Efron and R. Tibshirani, "Improvements on cross-validation: the 632+ bootstrap method," *Journal of the American Statistical Association*, vol. 92, no. 438, pp. 548–560, 1997.

[20] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on Software Engineering (TSE)*, vol. 25, no. 4, pp. 557–572, 1999.

[21] C. B. Seaman, F. Shull, M. Regardie, D. Elbert, R. L. Feldmann, Y. Guo, and S. Godfrey, "Defect categorization: making use of a decade of widely varying historical data," in *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*. ACM, 2008, pp. 149–157.

[22] T. Chen, W. Shang, J. Yang, A. E. Hassan, M. W. Godfrey, M. N. Nasser, and P. Flora, "An empirical study on the practice of maintaining object-relational mapping code in java systems," in *Proceedings of the 13th International Conference on Mining Software Repositories, MSR '16*, 2016, pp. 165–176.

[23] T.-H. Chen, M. Nagappan, E. Shihab, and A. E. Hassan, "An empirical study of dormant bugs," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR '14, 2014, pp. 82–91.

[24] G. Li, H. Zhu, T. Lu, X. Ding, and N. Gu, "Is it good to be like wikipedia?: Exploring the trade-offs of introducing collaborative editing model to Q&A sites," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ser. CSCW '15, 2015, pp. 1080–1091.

[25] A. Kittur and R. E. Kraut, "Harnessing the wisdom of crowds in wikipedia: Quality through coordination," in *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '08, 2008, pp. 37–46.

[26] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of stack overflow," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR '13, 2013, pp. 97–100.

[27] M. Duijn, A. Kučera, and A. Bacchelli, "Quality questions need quality code: Classifying code fragments on stack overflow," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR '15, 2015, pp. 410–413.

[28] F. Calefato, F. Lanubile, and N. Novielli, "How to ask for technical help? evidence-based guidelines for writing questions on stack overflow," *Inf. Softw. Technol.*, vol. 94, pp. 186–207, Feb. 2018.

[29] M. Hu, E.-P. Lim, A. Sun, H. W. Lauw, and B.-Q. Vuong, "Measuring article quality in wikipedia: Models and evaluation," in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, ser. CIKM '07, 2007, pp. 243–252.

[30] H. Cavusoglu, Z. Li, and K.-W. Huang, "Can gamification motivate voluntary contributions?: The case of stackoverflow Q&A community," in *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, ser. CSCW'15 Companion, 2015, pp. 171–174.

[31] P. B. Lowry, A. M. Curtis, and M. R. Lowry, "A Taxonomy of Collaborative Writing to Improve Empirical Research, Writing Practice, and Tool Development," *Journal of Business Communication*, vol. 41, no. 1, pp. 66–99, 2004.

[32] R. A. Calvo, S. T. O'Rourke, J. Jones, K. Yacef, and P. Reimann, "Collaborative writing support tools on the cloud," *IEEE Transactions on Learning Technologies*, vol. 4, no. 1, pp. 88–97, 2011.

[33] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in github: Transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, ser. CSCW '12, 2012, pp. 1277–1286.

[34] E. Kalliamvakou, D. Damian, K. Blincoe, L. Singer, and D. M. German, "Open source-style collaborative development practices in commercial projects using github," in *Proceedings of the 37th International Conference on Software Engineering - Volume 1*, ser. ICSE '15, 2015, pp. 574–585.

[35] C. Munteanu, R. Baecker, and G. Penn, "Collaborative editing for improved usefulness and usability of transcript-enhanced webcasts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08, 2008, pp. 373–382.

[36] M. M. Rahman and C. K. Roy, "An insight into the unresolved questions at stack overflow," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR '15, 2015, pp. 426–429.

[37] N. Hudson, P. K. Chilana, X. Guo, J. Day, and E. Liu, "Understanding triggers for clarification requests in community-based software help forums," in *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2015, pp. 189–193.

[38] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, and D. Fullerton, "Improving low quality stack overflow post detection," in *2014 IEEE International Conference on Software Maintenance and Evolution*, 2014, pp. 541–544.

[39] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Want a good answer? ask a good question first!" *arxiv*, 2013.

[40] ——, "Detecting high-quality posts in community question answering sites," *Inf. Sci.*, vol. 302, pp. 70–82, 2015.

[41] F. M. Harper, D. Raban, S. Rafaeli, and J. A. Konstan, "Predictors of answer quality in online Q&A sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08, 2008, pp. 865–874.

[42] L. Li, D. He, W. Jeng, S. Goodwin, and C. Zhang, "Answer quality characteristics and prediction on an academic q&a site: A case study on researchgate," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15 Companion, 2015, pp. 1453–1458.

[43] J. Antin and E. F. Churchill, "Badges in social media: A social psychological perspective," in *CHI 2011 Gamification Workshop Proceedings (Vancouver, BC, Canada, 2011)*, 2011.

[44] S. Wang, T.-H. Chen, and A. E. Hassan, "Understanding the factors for fast answers in technical q&a websites," pp. 1–42, 2017.

Shaowei Wang Shaowei Wang is a postdoctoral fellow in the Software Analysis and Intelligence Lab (SAIL) at Queen's University, Canada. He obtained his PhD from Singapore Management University, and BSc from Zhejiang University. His research interests include code mining and recommendation, software maintenance, developer forum analysis, and mining software repositories. More information at: https://sites.google.com/site/wswshaoweiwang/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2874470, IEEE Transactions on Software Engineering

Tse-Hsun (Peter) Chen Tse-Hsun (Peter) Chen is an Assistant Professor in the Department of Computer Science and Software Engineering at Concordia University, Montreal, Canada. He obtained his BSc from the University of British Columbia, and MSc and PhD from Queen's University. Besides his academic career, Dr. Chen also worked as a software performance engineer at BlackBerry for over four years. His research interests include performance engineering, database performance, program analysis, log analysis, and mining software repositories. Early tools developed by Dr. Chen were integrated into industrial practice for ensuring the quality of large-scale enterprise systems. More information at: http://petertsehsun.github.io/

Ahmed E. Hassan Ahmed E. Hassan is a Canada Research Chair in Software Analytics and the NSERC/Blackberry Industrial Research Chair with the School of Computing, Queen's University, Kingston, ON, Canada. His industrial experience includes helping architect the Blackberry wireless platform, and working for IBM Research at the Almaden Research Lab and the Computer Research Lab at Nortel Networks. Early tools and techniques developed by his team are already integrated into products used by millions of users worldwide. He is the named inventor of patents at several jurisdictions around the world including the United States, Europe, India, Canada, and Japan. Dr. Hassan serves on the editorial board of the IEEE Transactions on Software Engineering, the Journal of Empirical Software Engineering, and PeerJ Computer Science. He spearheaded the organization and creation of the Mining Software Repositories (MSR) conference and its research community.