Investigating Social Media in GitHub's Pull-Requests: A Case Study on Ruby on Rails

Yang Zhang, Gang Yin, Yue Yu, Huaimin Wang
National laboratory for Parallel and Distributed Processing, National University of Defense Technology
Changsha, 410073, China
zhangyanggfkd@gmail.com, jack_nudt@163.com, yuyue_whu@foxmail.com, whm_w@163.com

ABSTRACT

In GitHub, pull-request mechanism is an outstanding social development method by integrating with many social media. Many studies have explored that social media has an important effect on software development. @-mention as a typical social media, is a useful tool in social platform. In this paper, we made a quantitative analysis of @-mention in pull-requests of the project Ruby on Rails. First, we make a convictive statistics of the popularity of pull-request mechanism in GitHub. Then we investigate the current situation of @-mention in the Ruby on Rails. Our empirical analysis results find some insights of @-mention.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics – *Process metrics.*

General Terms

Experimentation, Human Factors.

Keywords

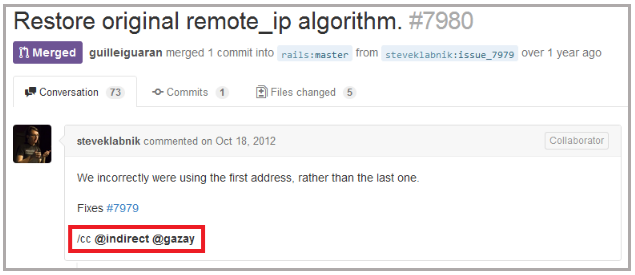
Social media; GitHub; pull-request; @-mention.

1. INTRODUCTION

The pull-request based software development is a new model for collaborating on distributed software development [1]. It makes more and more external developers contribute their code and suggestions to core developers. Compared with the traditional methods such as mailing list patching [2], the pull-request is a more efficient and socialized collaborative development model.

GitHub[1] is a social collaborative software development community. In addition to the generic tools such as automatic comparison of project branches, the platform integrates many social media tools involving follow [3], watch [3], comment action [4] and @-mention for contextual discussions and in-line code-reviews.

As shown in Fig.1, after a pull-request being sent to the source project, all developers in the GitHub have the chance to review it. They can freely communicate by submitting comments on the pull-request, the pull-request's diff patches or the pull-request's commits [4]. In particular, when they have doubts about this pull-request, they could reference an experienced developers for advice by simply placing a "@" symbol in front of the username they wish to reference [5].



guilleiguaran merged 1 commit into rails:master from steveklabnik:issue_7979 over 1 year ago

steveklabnik commented on Oct 18, 2012

We incorrectly were using the first address, rather than the last one.

Fixes #7979

/cc @indirect @gazay

Fig. 1. Developer uses @-mention to reference others

Previous work has identified the impact of @-mention on social platform such as Twitter[2]. These work found that the feature of @-mention enables users to directly reference others by putting a "@" symbol before their screen names [6]. @-mention is a strong predictor of information diffusion [7]. @-mention is a significant factor in enlarging the visibility of a post and helping initiate responses and conversations [8]. Basically, these previous researches focused on the correlation between the @-mention and the general social platform. Although we consider that the pull-request is popular in GitHub, we are not aware of any empirical study of @-mention dedicated in such claim. It inspires us to put forward an investigation of @-mention in the pull-requests.

In this paper, we conduct an investigation of the correlation between the @-mention and the pull-request in GitHub to find out what the impact of social media to the pull-requests is. First, in order to verify that the pull-request is popular in GitHub, we obtain some insights of the popularity of the pull-request in GitHub by using the statistical approaches. Then we perform a detailed investigation of the @-mention in pull-requests on a famous project called Ruby on Rails. Our results give an explicit description of the current situation of @-mention and elicit some important implications for the developers to make better use of @-mention in GitHub.

The remainder of this paper is structured as follows. Section 2 describes related concepts. In Section 3, we introduce our research questions and our empirical study methodology, and Section 4 presents results of the study. Related work and threats to validity are discussed in Section 5 and Section 6. We conclude the article in Section 7.

2. PRELIMINARIES

In this section, we give a brief introduction of pull-request and @-mention.

2.1 Pull-Requests

In GitHub, the developers fork the project's main repository and make their own changes. When they think it is ready to submit these changes to the main repository, they create a pull-request to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CrowdSoft'14, November 17, 2014, Hong Kong, China
Copyright 2014 ACM 978-1-4503-3224-8/14/11...$15.00
http://dx.doi.org/10.1145/2666539.2666572

[1] Https://github.com/

specify a local branch to be merged with a branch in the main repository. Then, one of the internal contributors of the project inspects the changes and pulls them to the project's master branch. If the pull-request does not meet the standard or needs further improvement, the submitter of the pull-request would update his pull-request by attaching some new commits. During such processing, all the contributors can review and discuss in the pull-request until it be closed.

2.2 @-Mention

In recent year, more and more online social platforms, such as Facebook[2], Twitter[3], use @ to denote a reference or a reply. The feature of @-mention enables users to directly reference others by putting a "@" symbol before their username. Then @-mention can automatically interpret these as links to the user's profile. We find that in the pull-requests of GitHub, @-mention can be found in the pull-request' title, pull-request's description body (pull-request's body) and pull-request's comments. In reality, @-mention in the pull-request's title does not have the link function because it is just a text. So we only discuss the @-mention that used in the pull-request's body and the pull-request's comments in our investigation.

3. METHODOLOGY

In this section, we give our research questions that we focus on in our investigation and introduce our datasets and the preprocessing.

3.1 Research Questions

In order to have a detailed research, we give the following research questions to explore and assess the correlation between the @-mention and the pull-request:

RQ1: What is the popularity of pull-requests in GitHub?

In answer to this research questions, we study the pull-request related events and investigate the basic distribution of pull-requests based on more than 10 million data from two famous datasets.

RQ2: What is the current situation of @-mention used in the pull-request paradigm?

For answering this research question, we investigate the distribution of @-mention in pull-requests and analysis the difference between the pull-requests with @-mention and the pull-requests without @-mention on a famous project called Ruby on Rails.

3.2 Datasets

In our investigation, we choose two famous datasets, GitHub Archive and GHTorrent, to build our research datasets. GitHub Archive[4] is a project that maintains logs of significant actions on the Git repositories stored on GitHub. As shown in Table I, the development activities in GitHub are aggregated in hourly archives. The Archive provides 18 types of events, such as new commits, fork, commenting, adding members etc. The Archive encodes these events into a Json file. During our work, we download and parse the Archive data from January 2013 to March 2014. The full volume of these Archive data is approximately 160GB. There are over 100 million events. GHTorrent[5] is a scalable, offline mirror of the data offered through the GitHub Rest API [9]. In the GHTorrent, there are already have more than 2TB data dumps of both its raw data that stored in MongoDB, and more than 20GB metadata that stored in MySQL [10]. As shown in Table II, after downloading and parsing the database dump, we can get 20 types of data, such as commits, issues, pull-requests, comments, projects, which are almost all of the development information of the repositories in the GitHub.

TABLE I. 18 TYPES OF EVENTS IN GITHUB ARCHIVE

Events	CommitCommentEvent, CreateEvent, DeleteEvent, DownloadEvent, FollowEvent, ForkEvent, ForkApplyEvent, GistEvent, GollumEvent,IssueCommentEvent, IssuesEvent, MemberEvent, PublicEvent, PullRequestEvent, PushEvent, ReleaseEvent, PullRequestReviewCommentEvent, WatchEvent

TABLE II. 20 TYPES OF DATA IN GHTORRENT

Data	commit_comments, commit_parents, commits, followers, issue_comments, issue_events, issue_labels, issues, organization_members, project_commits, project_members, projects, pull_request_comments, pull_request_commits, pull_request_history, pull_requests, repo_labels, repo_milestones, users, watchers

3.3 Preprocessing

During the preprocessing of our dataset, we need to extract the information of @-mention from the pull-request's comments. As shown in Fig.2, the extraction work can be divided into 4 steps. First step, we extract the pull-request's body and comments information by scanning the unique pull-request ID. We define these information as the pull-request's text. Second step, we use the textual analysis to judge whether the pull-request's text contains "@" symbol. If the text contains "@", then we go to third step, otherwise we scan the next pull-request's text. Third step, we query the dataset to judge whether the "@" is a valid @-mention operation. Because some text in back of @ are not real username in GitHub. If it is a valid @-mention operation, then we go to the fourth step, otherwise we scan the next pull-request's text. Fourth step, we insert the valid @-mention information into our MySQL dataset for further statistics and investigation.

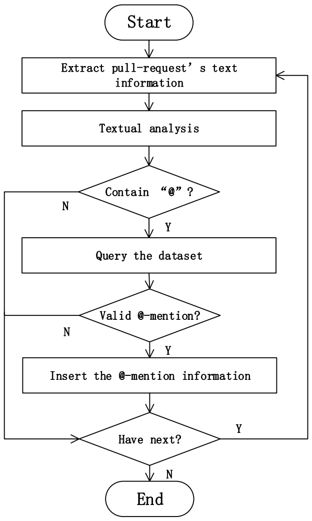


Fig. 2. The process of extracting @-mention information

[2] Https://www.facebook.com/

[3] Https://www.twitter.com/

[4] Http://www.githubarchive.org/

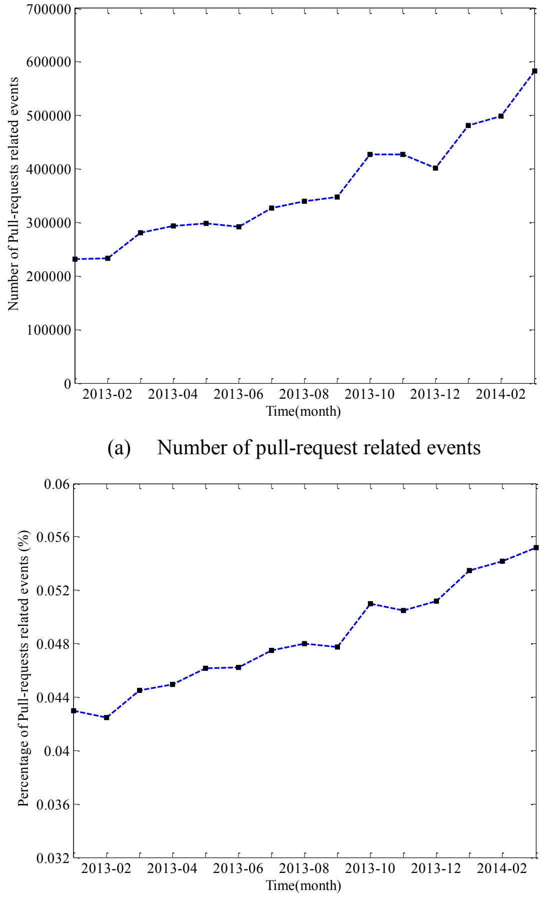
[5] Http://ghtorrent.org/

4. INVESTIGATION RESULTS

In this section, we present the results of our investigation. These results are reported as responses to the research questions that were provided in Section 3.1.

4.1 RQ1: Popularity of Pull-Requests

In GitHub Archive, from January 2013 to March 2014, the absolute number of pull-request related events (PullRequestEvent and PullRequestReviewCommentEvent) goes up steadily to the highest point 583239 per month, as shown in Fig.3(a). In addition to the number increasing, as shown in Fig.3(b), the percentage of pull-requests related events is moderate increasing, reaching the highest point 5.5%.

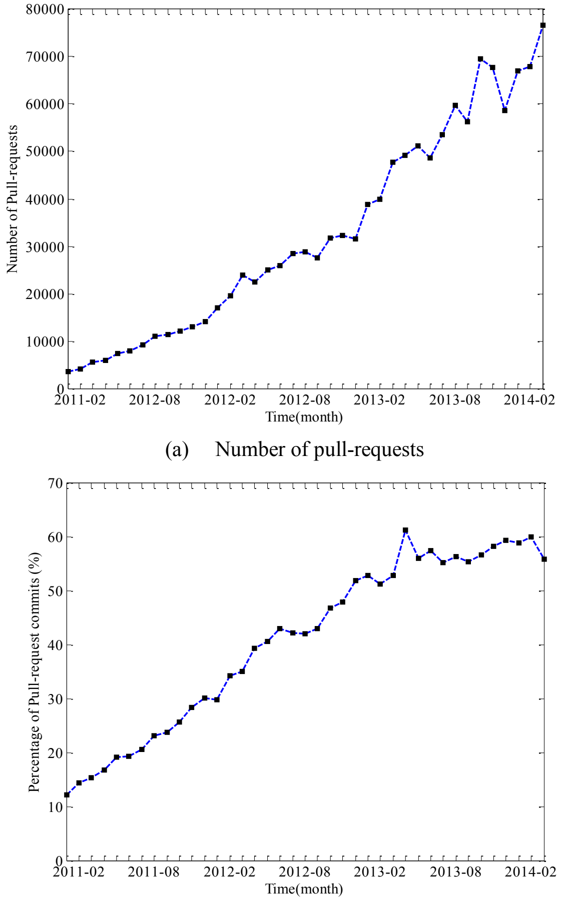


(a) Number of pull-request related events

(b) Percentage of pull-request related events

Fig. 3. The growth of pull-request related events

Furthermore, we investigate the quantity of the pull-requests and the percentage of pull-request commits. For avoiding disturbances, we mainly analyze the data of 3587 projects which receive at least 100 pull-requests and not be deleted from the latest database dump. According to our monthly statistics, from June 2011 to March 2014, the absolute number of new pull-requests is increasing dramatically with a few moderate fluctuations. The highest point is 62119 new pull-requests per month as shown in Fig.4(a). The percentage of pull-requests' commits is increasing rapidly to the highest point of 61% and then fluctuated within the range of 55% to 60% as shown in Fig.4(b). In recent months, nearly 60% commits were transmitted by the pull-request mechanism compared to the traditional share repository approach. This indicates that ***pull-request mechanism is a popular model in GitHub.***



(a) Number of pull-requests

(b) Percentage of pull-request commits

Fig. 4. The growth of pull-requests and pull-request commits

4.2 RQ2: Current Situation of @-Mention

For analyzing the current situation of *@-mention*, we do a case study on the project called Ruby on Rails. As shown in Table III, Ruby on Rails is a famous project which is maintained in GitHub during its whole development.

TABLE III. BASIC INFORMATION OF RUBY ON RAILS

language	stars	forks	pull-requests	commits	contri-butions
Ruby	21772	7980	9129	43526	2283

In the total development period of Ruby on Rails, we investigate the utilization of *@-mention* in pull-requests. After filtering the 7928 closed pull-requests from Ruby on Rails, in Fig.5, we find that the utilization of *@-mention* is probably presents four stages: 1) in the early stage, from September 2010 to January 2011, the Ruby on Rails just starts developing with pull-requests, so the utilization of *@-mention* is very low; 2) in the rapid developing stage, from January 2011 to May 2012, the development activities of Ruby on Rails is rapidly growing, the utilization of *@-mention* is fast increasing too; 3) in the stable developing stage, from May 2012 to July 2013, Ruby on Rails is in a steady phase of development, the utilization of *@-mention* is basically unchanged at 45%; 4) in the mature developing stage, from July 2013 to March 2014, with the Ruby on Rails is more mature and stable, the number of development activities for new functions or fixing bugs is decreasing, so the utilization of *@-mention* is generally decreasing too.

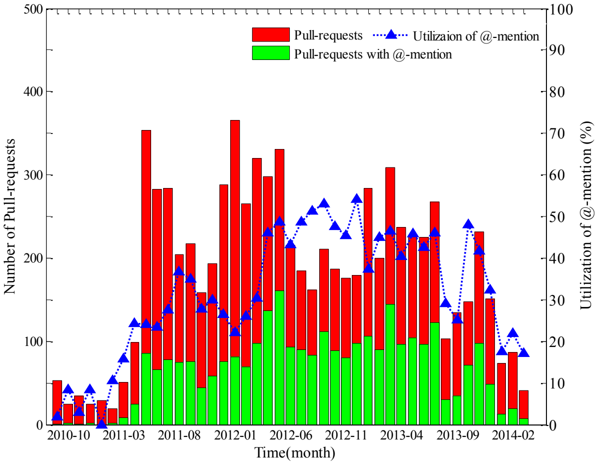


Fig. 5. The utilization of *@-mention* in Ruby on Rails

In Section 3.1, we show that @-mention is usually used in the pull-request's body and pull-request's comments. As shown in table IV, 27.6% @-mention come from the pull-requests' bodies. 72.4% @-mention come from the pull-requests' comments (issue comments: 91.7%, pull-request review comments: 8.1% and commit comments: 0.2%). This indicates that ***in Ruby on Rails, @-mention are mainly used in the pull-request's body and the issue comments***.

TABLE IV. PERCENTAGE OF @-MENTION IN DIFFERENT PLACES

Location	Percentage
body	***27.6%***
comments	***72.4%***
pull-request review comments	8.1%
commit comments	0.6%
issue comments	91.3%

TABLE V. UTILIZATION OF @-MENTION IN COMMENTS AND BODY

	Total	
	#PR	Ratio
have comments	4928	62.2%
@ in comments	1360	27.6%
have body	6767	85.4%
@ in body	812	12.0%

As shown in table V, in the 7928 closed pull-requests, there are 4928 (62.2%) pull-requests that have comments and 6767 (85.4%) pull-requests that have body. The percentage of *@-mention* used in body is 12.0%, while the value in comments is 58.2% and the value in total is 42.2%. 27.6% of these *@-mention* appear in pull-requests' body and 72.4% appear in comments (91.7% appear in the issue comments). This indicates that ***in Ruby on Rails, @-mention is not widely used in the pull-requests and @-mention is more likely to be used in comments than the pull-request's body***.

From our statistics, in Ruby on Rails, the pull-requests with *@-mention* have 3.9 commits (median: 1.0), 6.7 comments (median: 4.0) and 3.3 participants (median: 3.0). In the opposite, the pull-requests without *@-mention* have 5.5 commits (median: 1.0), 1.0 comments (median: 0.0) and 1.5 participants (median: 1.0). We find that ***in Ruby on Rails, the pull-requests with @-mention are more complex than the pull-requests without @-mention.***

5. RELATED WORK

In previous work, Riemer K and Richter A [11] find that decision makers should vest trust in their employees when they put microblogging to productive use in their group work environments. Louridas P [12] find that wikis can be used to support defect tracking, documentation, requirements tracking, test case management as well as the creation of project portals. Ahmadi et al. [13] find that in today, developers frequently makes use of social media to augment tools in their development environments. Park S et al. [14] find that blogs can be frequently used by developers to discuss the release of new features and to support requirements engineering. O'reilly T [15] find that social media tools supports crowdsourcing as well as a many-to-many broadcast mechanism. Storey M A et al. [16] investigated using social media in software development at the team, project and community levels involving its benefits, risks and limitations. Julia Kotlarsky et al. [17] find that human-related issues involving rapport and transactive memory were important for collaborative work in the software development. Black S et al. [18] find that social media can enable better communication through the software system development process.

6. THREATS TO VALIDITY

Our statistical analysis uses the number of commits etc. as measurements to verify the characteristics of pull-requests with @-mention. Future work is needed on analyzing the total handling time of a pull-request. Because the total handling time is better reflecting the complexity of the pull-request. In the future, we would discuss the difference of total handling time between the pull-requests with @-mention and without @-mention. In reality, there are many cost time in the processing of a pull-request, such as the delay time before the first comment and the delay time among the comments. Future work should investigate the impact of @-mention on these cost time. Also we consider that the different location of @-mention might have different influence on the processing of pull-request. In the future, we would investigate the correlation between the location of @-mention and the processing of pull-request.

7. CONCLUSIONS

This investigation obtains some primary understanding of @-mention in the GitHub's pull-requests, including the popularity of pull-request and the basic current situation of @-mention. The statistics results indicates that pull-request mechanism is a popular model in GitHub. By doing a case study on Ruby on Rails, we find that the @-mention are mainly used in the pull-request's body and issue comments. @-mention is not widely used and @-mention is more likely to be used in comments than the pull-request's body. But the pull-requests with @-mention are more complex than the pull-requests without @-mention. There are still some unknown information about @-mention in the pull-requests. More detailed researches should be conducted to make a better use of @-mention in the pull-request based software development. In the future, we would do some in-depth analysis of @-mention to help the researchers and developers understand the significance of @-mention in the pull-requests well.

8. ACKNOWLEDGMENT

The research is supported by the National High Technology Research and Development Program of China (Grant No. 2012AA011201), and the National Natural Science Foundation of China (Grant No. 61432020 and 61472430).

9. REFERENCES

[1] Gousios G, Pinzger M, and van Deursen A, "An exploration of the pull-based software development model," in Proc. of the 36th International Conference on Software Engineering (ICSE), 2014.

[2] Bird C, Gourley A, Devanbu P, et al. Mining email social networks[C]//Proceedings of the 2006 international workshop on Mining software repositories. ACM, 2006, pp. 137-143.

[3] Tsay J, Dabbish L, and Herbsleb J D, "Social media in transparent work environments," in Proc. of the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2013, pp. 65-72.

[4] Dabbish L, Stuart C, Tsay J, and Herbsleb J, "Social coding in GitHub: transparency and collaboration in an open software repository," in Proc. of the ACM 2012 conference on Computer Supported Cooperative Work, 2012, pp. 1277-1286.

[5] Meeder B, Tam J, Kelley P G, and Cranor L F, "RT@ IWantPrivacy: Widespread violation of privacy settings in the Twitter social network," in Proc. of the Web, vol. 2, 2010.

[6] Storey M A, Treude C, van Deursen A, and Cheng L T, "The impact of social media on software engineering practices and tools," in Proc. of the FSE/SDP workshop on Future of software engineering research, 2010, pp. 359-364.

[7] Begel A, DeLine R, Zimmermann T. Social media for software engineering[C]//Proceedings of the FSE/SDP workshop on Future of software engineering research. ACM, 2010, pp. 33-38.

[8] Black S, Harrison R, and Baldwin M, "A survey of social media use in software systems development," in Proc. of the 1st Workshop on Web 2.0 for Software Engineering, 2010, pp. 1-5.

[9] Gousios G, Spinellis D. GHTorrent: Github's data from a firehose[C]//Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on. IEEE, 2012: 12-21.

[10] Gousios G, Vasilescu B, Serebrenik A, and Zaidman A, "Lean GHTorrent: GitHub data on demand," in Proc. of the 11th Working Conference on Mining Software Repositories (MSR), 2014, pp. 384-387.

[11] Riemer K, Richter A. Tweet inside: Microblogging in a corporate context[J]. Proceedings of the 23rd Bled eConference, 2010: 1-17.

[12] Louridas P. Using wikis in software development[J]. Software, IEEE, 2006, 23(2): 88-91.

[13] Ahmadi N, Jazayeri M, Lelli F, et al. A survey of social software engineering[C]//Automated Software Engineering-Workshops, 2008. ASE Workshops 2008. 23rd IEEE/ACM International Conference on. IEEE, 2008: 1-12.

[14] Park S, Maurer F. The role of blogging in generating a software product vision[C]//Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. IEEE Computer Society, 2009: 74-77.

[15] O'reilly T. What is Web 2.0: Design patterns and business models for the next generation of software[J]. Communications & strategies, 2007 (65).

[16] Storey M A, Treude C, van Deursen A, et al. The impact of social media on software engineering practices and tools[C]//Proceedings of the FSE/SDP workshop on Future of software engineering research. ACM, 2010: 359-364.

[17] Kotlarsky J, Oshri I. Social ties, knowledge sharing and successful collaboration in globally distributed system development projects[J]. European Journal of Information Systems, 2005, 14(1): 37-48.

[18] Black S, Harrison R, Baldwin M. A survey of social media use in software systems development[C]//Proceedings of the 1st Workshop on Web 2.0 for Software Engineering. ACM, 2010: 1-5.