

A Hybrid Approach for Mining the Weakness of API Documentation

Fan Qiang^{1,2*}, Wang Tao^{1,2}, Yin Gang^{1,2}, Yu Yue^{1,2} & Wang Huaimin^{1,2}

¹Key Laboratory of Parallel and Distributed Computing;

²College of Computer, National University of Defense Technology, Changsha, 410073, China

Citation Fan Q, Wang T, Yin G, et al. A Hybrid Approach for Mining the Weakness of API Documentation. Sci China Inf Sci, for review

Dear editor,

API documentation plays an important role in software development and reuse [1] for both API maintainers and API users. Well-written documentation will help developers effectively understand and reused code [2], and reduce their time to focus on desired interfaces and functions instead of the entire system [3]. Most of the high-quality open source projects maintain a complete and informative official documentation. The API documentation typically conveys detailed specifications such as class/interface hierarchies and method descriptions which can benefit developers much [4]. However, despite of its authoritativeness and thoroughness, the single-sourced official documentation does not always meet the developers' requirements [5]. As of January 2018, over 1 million questions about Android are posted to StackOverflow, and about 16% of them are related to API (by searching API and Android in StackOverflow). The missing of important information in API documentation makes developers spend much time on searching and seeking for help. During this time, developers will obtain a lot of unconfirmed information, which can affect the quality of the project and the developers' understanding of the API.

This research studied the weakness of API documentation based on a hybrid approach (including manual inspection and online survey), which can be used to improve the quality of API documentation. We explore the crowd discussions about An-

droid API in Stack Overflow, compare them with the official documentation and identify what questions had troubled the third-party developers and why they are confused even there are official documentation. In addition, we conducted an extensive online survey on these developers who involved in the discussions to understand their trouble, which mutual verifies our discovery. Finally, an empirical study of 1000 questions and 319 questionnaire responses was conducted, and we identify 10 weaknesses of API documentation.

API documentation overview. API documentation is a technical content deliverable, most of time containing details about the functions, classes, return types and arguments, etc. Most of time, official API documentation is maintained by the core team or enthusiastic fans of the project, and a well flowed documentation is always well edited and with a good structure. Under ideal conditions, a good API documentation should provide a concise and basic information about the API, which can put across how to effectively use and integrate with an API to API users. However, the different perspectives between API developers and users make it a challenge. On the one hand, The difference, such as shared keywords and technical background, makes API users hard to use the API, which increases its costs. On the other hand, it also makes API developers hard to figure out API users requirements.

Facing the weak of API documentation, API

* Corresponding author (email: fanqiang09@nudt.edu.cn)

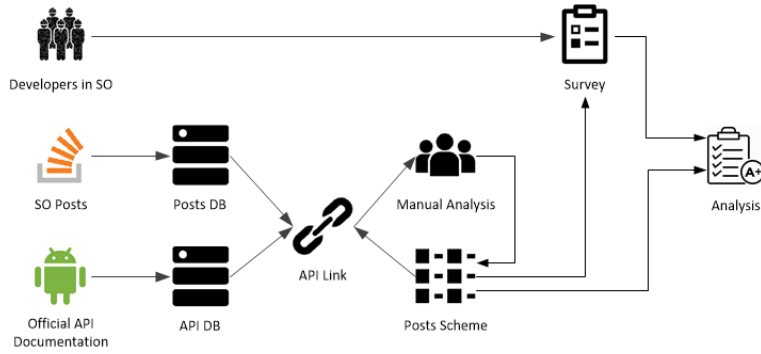


Figure 1 The process of the hybrid approach

users often access to Q&A websites like StackOverflow to seek help from other developers. They post their question or problem they meet in Q&A websites, and other developers who have solved the problem will provide help to them and answer the post. As time goes on, these websites accumulate tremendous amount of discussion about API, which can be seen as a kind of crowd documentation in [5, 7, 8]. Compared with official API documentation, the crowd documentation is more dynamic and interactive. API users can search in historic discussions to find solution, or post a new question to gain solution from other developers. The advantages of crowd documentation are that API users can find new solutions from other developers' experience, and the question oriented way can let API users locate information more quickly.

The flexibility and question-oriented way make more and more API users access crowd documentation to find solution, but there are still some limitations. For instance, the answers from crowd developers are uneven, and many answers may be not well edited and proofread. So it is hard to guarantee correctness of information provided by crowds. The process of filtering out the right answer is time consuming and upsetting. Besides, crowd developers need time to figure out the question and organize language answering it. Some questions may not get attention, and few developers will answer or response to it. Factors like these result that API users can not receive help in time. So a well edited API documentation plays an irreplaceable role in providing reliable information and swift inquiry. It is necessary to figure out the weakness of API documentation and try to improve it.

Approach. In this research, we propose a hybrid approach to explore the requirements of the third-party developers about API documentation. The approach includes manual inspection and online survey, and the overview of our approach is

shown in Figure 1.

Firstly, we link Android API and Android-related posts, which is used to filter out posts discussing Android API. To extract posts discussed about Android API, we used the process adopted from researches [9]. In our case, a link is a connection between a StackOverflow question and an Android class (or method). By using our method, we finally filtered 54,927 Stack Overflow questions, which link to at least one Android API class or method.

Secondly, we use manual inspection to categorize the topics and causes of questions based on posts we select in process of linking API. In addition, we identify information provided in official documentation, which is used to compare different perspective of API developers and users. We randomly select 100 posts associated with Android API, and inspect the questions and answers of the posts to find what weakness of API documentation result in this question. Through a rigorous analysis of existing literature and our own experience with working and analyzing, we identify 10 causes of questions. after that, 10 master students repeat the previous process from new selected questions to exam the stability of the taxonomy definition, and each selected post will be assigned to two students. We use our stabilized taxonomy to identify 1000 Android-related questions.

Then, we use the results of manual inspection to design a research survey, and deliver to users in Stack Overflow. We release the questionnaires on SurveyMonkey¹⁾, one of the most famous web services for online surveys. The questionnaires contain questions like "What is the weakness of API documentation result that you seek help from StackOverflow". We send the survey questionnaires to 1) the users who have ever asked questions about Android API usage and 2) the users who have ever answered questions about Android API usage. Within a period of 7 days, we have

1) <https://www.surveymonkey.com>

received overall 319 responses in total.

Result. From our hybrid approach, we find that both manual inspection and online surveys present a similar result. Next, we will list some key conclusions of our research:

Nearly 10% API users never use documentation. They believe that searching answers or asking questions in the Stack Overflow is more efficient than searching answers in the documentation. In addition, Stack Overflow is more friendly to locate what they want.

Lack of examples is the most important factor that caused many questions for the third-part developers. Nearly 50% of questions from manual inspection and 50% of third-part developers from survey show that lacking examples causes problems. Most of time, the official documentation provides very detailed explanation of the API about what the API used for and means of each parameter. These information is useful for third-part developers to search the API, but lacking examples makes them have to search extra information to figure out how to use the API. Furthermore, lacking examples results in many wrong usage of the API, which makes programs crash and run error.

Searching API is one of main requirements of API users. Nearly 20% of questions from manual inspection and 30% of third-part developers from survey show that searching API is important. In the result of manual inspection, the number of pageviews about finding API is more than other types of questions, which also verifies our conclusion.

Lack of precautions causes API used wrong. Over 30% of questions and nearly 20% of third-part developers show precautions is important. Precautions can prevent wrong usage of API. Lacking precautions costs API users much time to fix wrong usage of their program.

The content of the official documentation is far from complete. Over 30% of questions from manual inspection and over 30% of third-part developers from survey shows incomplete of the documentation. They can not always find what they want from the documentation. They have to seek help from other developers to find information they want and better understand the API.

Locating wanted information in documentation is hard, which disturbs the third-part developers. Over 30% of questions and nearly 30% of API users shows the problem. More and more information are added in documentation with version iterations, which increases the cost of locating information and reduces the efficiency of development. Good searching mechanism and recommendation system are helpful for them to use docu-

mentation.

Sometimes the statement of documentation are hard to understand. Nearly 20% of questions and 30% of third-part developers shows the problem. The official documentation are edited by manufacturers with accurate and concise expression, which may be Incomprehensible for them.

Acknowledgements This research is supported by National Natural Science Foundation of China (Grant No.61432020, 61472430, 61502512 and 61303064) and National Key R&D Program of China (2016-YFB1000805).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

- 1 R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry et al., Bioconductor: open software development for computational biology and bioinformatics, *Genome biology*, vol. 5, no. 10, p. R80, 2004
- 2 A. Forward and T. C. Lethbridge, The relevance of software documentation, tools and technologies: a survey, in *Proceedings of the 2002 ACM symposium on Document engineering*. ACM, 2002, pp. 2633.
- 3 T. Roehm, R. Tiarks, R. Koschke, and W. Maalej, How do professional developers comprehend software? in *Proceedings of the 34th International Conference on Software Engineering*. IEEE Press, 2012, pp. 255265.
- 4 U. Dekel and J. D. Herbsleb, Improving api documentation usability with knowledge pushing, in *Proceedings of the 31st International Conference on Software Engineering*. IEEE Computer Society, 2009, pp. 320330.
- 5 C. Chen and K. Zhang, Who asked what: Integrating crowdsourced faqs into api documentation, in *Companion Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 456459.
- 6 G. Uddin and M. P. Robillard, How api documentation fails, *IEEE Software*, vol. 32, no. 4, pp. 6875, 2015.
- 7 J. Brandt, M. Dontcheva, M. Weskamp, and S. R. Klemmer, Examplecentric programming: integrating web search into the development environment, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 513522.
- 8 B. Hartmann, M. Dhillon, and M. K. Chan, Hyper-source: bridging the gap between source and code-related web sites, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 22072210.
- 9 M. Linares-Vasquez, G. Bavota, M. Di Penta, R. Oliveto, and D. Poshyvanyk, How do api changes trigger stack overflow discussions? a study on the android sdk, in *proceedings of the 22nd International Conference on Program Comprehension*. ACM, 2014, pp. 8394.