

An Empirical Study of Obsolete Knowledge on Stack Overflow

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

Abstract—Stack Overflow accumulates an enormous amount of software engineering knowledge. However, as time passes, certain knowledge in answers may become obsolete. Such obsolete answers, if not identified or documented clearly, may mislead answer seekers and cause unexpected problems (e.g., using an out-dated security protocol). In this paper, we investigate how the knowledge in answers becomes obsolete and identify the characteristics of such obsolete answers. We find that: 1) More than half of the obsolete answers (58.4%) were probably already obsolete when they were first posted. 2) When an obsolete answer is observed, only a small proportion (23.5%) of such answers are ever updated. 3) Answers to questions in certain tags (e.g., tags related to web and mobile development) are more likely to become obsolete. Our findings suggest that Stack Overflow should develop mechanisms to encourage the whole community to maintain answers (to avoid obsolete answers) and answer seekers are encouraged to carefully go through all information (e.g., comments) in answer threads.

Index Terms—Q&A Website, Stack Overflow, Obsolete Knowledge, Knowledge Sharing.

1 INTRODUCTION

TECHNICAL Q&A websites are becoming an important and popular platform for knowledge sharing and learning. They have revolutionized how users seek knowledge on the Internet. When users face unsolvable problems, they often try to search for solutions via search engines (e.g., Google). A case study shows that Google developers perform an average of 12 code search queries per weekday [1]. Search engines commonly direct users to technical Q&A websites in response to their queries. As a prominent example, Stack Overflow, one of the most popular Q&A websites for users, has collected an enormous amount of knowledge, which includes 15 million questions, 23 million answers, and 62 million comments as of September 2017¹.

Software systems evolve at a rapid pace nowadays. For instance, Android has released 15 major versions and 51 minor versions since September 2008 [2]. Android is evolving at a rate of 115 API updates per month on average according to a study by McDonnell et al. [3]. Such rapid evolution may make the knowledge in some Stack Overflow answers obsolete over time. Fig. 1 presents an example of such a case, where the user was directed from Google to a Stack Overflow answer. However, the user found that the content of the answer thread (including the answer and the discussions in the comments) was obsolete and asked whether Stack Overflow has any mechanisms to handle such a situation².

- H. Zhang, S. Wang and A. E. Hassan are with the Software Analysis and Intelligence Lab (SAIL), Queens University, Kingston, Ontario, Canada. E-mail: hzhang,shaowei,ahmed@cs.queensu.ca
- T. Chen is with the Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada. E-mail: peterc@encs.concordia.ca
- Y. Zou is with the Department of Electrical and Computer Engineering, Queens University, Kingston, Ontario, Canada. E-mail: ying.zou@queensu.ca
- Shaowei Wang is the corresponding author.

1. <https://data.stackexchange.com/stackoverflow/>
2. <https://stackoverflow.com/questions/1447208/>

Additionally, a survey of 453 Stack Overflow users reports that outdated code on Stack Overflow is one of the most important issues that users complain about [4].



Fig. 1: An example of a user complaining in a comment that the Stack Overflow answer thread (including the answer and the discussions in the comments) is obsolete.

Obsolete answers are detrimental to answer seekers. For example, a user found a piece of code that matches his/her needs and reused it in his/her own project. However, the user may not realize that the used APIs in the code are obsolete. Using such obsolete APIs could potentially result in software quality problems (e.g., using an outdated security framework API), and may increase maintenance difficulties. Therefore, it is necessary to provide insights on how to track or alleviate this problem.

In this paper, we study 52,177 Stack Overflow answer threads (each answer thread includes all answers to a question (i.e., accepted & non-accepted answers) and all the comments that are associated with them) to understand how the knowledge that is embedded in answer threads becomes obsolete and the characteristics of such obsolete answers. We also perform a qualitative study to understand the evidence that users provide to support their obsolete observations and the activities that users perform after an

answer is observed as obsolete. We structure our study by answering the following research questions:

- **RQ1: What happens when an answer is observed as obsolete?**

More than half of the studied obsolescence observations refer to answers that were probably already obsolete when they were first posted. Most users did not update obsolete answers or add new answers to address the obsolescence. On average, it took 101 days for users to react to an observed obsolete answer.

- **RQ2: Whether answers to questions that are associated with particular tags are more likely to become obsolete?**

Answers to questions that are associated with certain tags (e.g., web development and mobile development related tags) are more likely to become obsolete. The majority of the answers become obsolete due to the evolution of their associated programming language, outdated references and tool updates. Therefore, users need to pay more attention to such answers when looking for answers on Stack Overflow.

- **RQ3: Who observes obsolete answers and what evidence do they provide?**

The majority of the obsolete answers were not observed by the original answerers. Also, most obsolescence observations are supported by evidence (e.g., updated information, a version information, or a reference).

Based on our observations, we suggest that Stack Overflow should develop mechanisms (e.g., rewarding badges or reputation scores) to encourage the whole community to maintain answers and flag obsolete answers. Answerers are encouraged to include information of the valid version or time of the knowledge if possible when contributing answers. Answer seekers are encouraged to carefully go through the comments that are associated with answers in case the obsolescence of an answer is noted in the comments, especially for the answers in questions that are related to particular tags (e.g., web and mobile development related tags). We also shared our findings with Stack Overflow and Stack Overflow developers concurred with our findings.

Paper Organization: The rest of the paper is organized as follows. Section 2 presents the background. Section 3 introduces our data collection process. Section 4 presents the results of our research questions. Section 5 discusses the implications of our study. Section 6 presents the potential threats to the validity of our observations. Section 7 discusses related work. Finally, Section 8 concludes the paper.

2 BACKGROUND

In this section, we briefly introduce the mechanism of question answering and commenting on Stack Overflow. We also discuss how answers on Stack Overflow become obsolete.

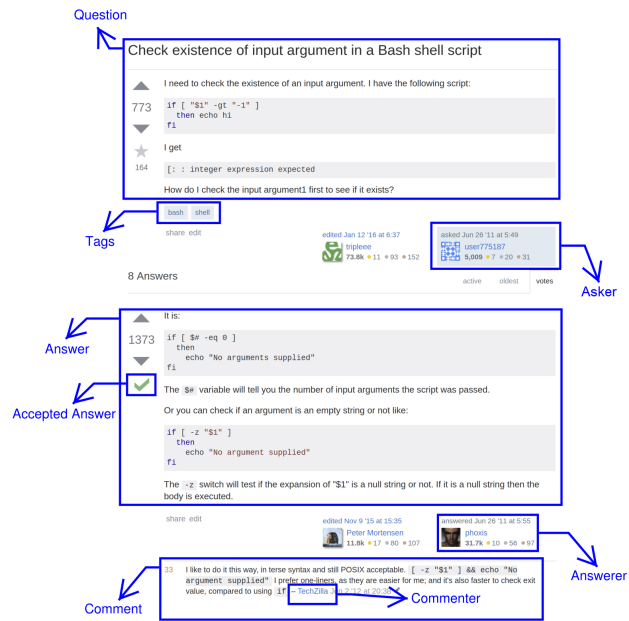


Fig. 2: An example of a Stack Overflow question, its accepted answer and the associated comments to its accepted answer.

2.1 The Question and Answer Process on Stack Overflow

Stack Overflow provides a platform for the asking and answering of questions. Askers post questions which include a textual description on Stack Overflow. Askers can include code snippets and other references (e.g., URLs or images) to enrich their posted question. Answerers respond to posted questions based on their experience and expertise. Each question may receive multiple answers from different answerers. However, at most one answer could be accepted by the asker as the *accepted answer* 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. Up votes reflect positive feedback from the community and down votes reflect negative feedback. A snapshot of a question and its accepted answers is shown in Fig. 2.

In the rest of the paper, we refer to a question, its corresponding answers (i.e., both accepted and non-accepted answers) and all the associated comments with these answers together as a *question thread*. We refer to an answer (could either be accepted or non-accepted answers) and its comments as an *answer thread*.

Users tag questions³ into well-defined categories. Tags are located under the content of a question to capture the topics with which the question is associated. Each question can have at most five tags and must have at least one tag. Askers need to specify the tags when they create a question. For example, the question in Fig. 2 is associated with the tags "bash" and "shell". In the rest of the paper, we say that an answer is associated with a particular tag if the answer belongs to a question that is associated to that tag. In RQ2,

3. <https://stackoverflow.com/help/tagging>

25 I agree with all three of your reasons which is why I include jquery from Google on my production sites. Instead of the the js dynamic injection for SSL pages I just reference url in a script tag without the protocol specified. Seems to work fine for me. `<script src="//ajax.google..."></script>` – Aaron Wagner Feb 13 '09 at 23:59

1 Interesting idea... But if you're going to use DNS poisoning to hijack the JQuery load why not just hijack the the whole site request? Or how about the Google Analytics script? – Dscoduc Feb 16 '09 at 8:04

show 4 more comments

appears after clicking



@vol7ron: In case you use `//googleepis...`, you don't need to worry about choosing between http and https - the browser does it for you. That makes the whole `document.write()` part obsolete. – SF. Feb 9 '16 at 10:14

Fig. 3: An example of answers with more than 5 comments. Note the comment that observed the obsolescence in the answer was a hidden comment and users had to click “show 4 more comments” in order to notice it.

we study whether answers to questions that are associated to particular tags are more likely to become obsolete.

2.2 Commenting Mechanism on Stack Overflow

Stack Overflow allows users to leave comments for each question and for each associated answer. Comments may not be easily accessible to users because once the number of comments reaches a certain amount (i.e., 5), additional comments are hidden. This folding is implemented to help improve the UI presentation of threads (i.e., folded by the user interface) by default. Only the top 5 most voted comments are shown. For instance, the comment shown in Fig. 3 is a comment which points out that `document.write()` is obsolete; however, this comment is hidden by default⁴. If a user does not click “show 4 more comments”, he would not recognize the obsolescence of the answer. In addition, such hidden comments are not indexed by Google⁵.

2.3 Obsolete Answers on Stack Overflow

As we noted in Section 1, Stack Overflow users complain about the obsolescence of answers. There are various reasons that an answer could become obsolete on Stack Overflow. For instance, APIs could become deprecated when a new API version is released. Additional documents may be provided by the company to make a smooth transition between API versions. However, if a user is not actively monitoring the API version changes, it may not be clear to him/her when one version becomes obsolete, the reason for it becoming obsolete, and what to do in response to the API's obsolescence. Obsolete answers are problematic on Stack Overflow. However, there exists no mechanisms in place today to alleviate the problem of obsolete answers. Thus, in this paper, we would like to closely examine the obsolescence of answers in an effort to propose ways to help Stack Overflow deal with it in an effective & efficient manner. To do so, we investigate what happens once someone identifies that an answer has become obsolete and whether answers in questions that are associated with particular tags

4. <https://stackoverflow.com/questions/547482/>

5. <https://meta.stackexchange.com/questions/304904/>

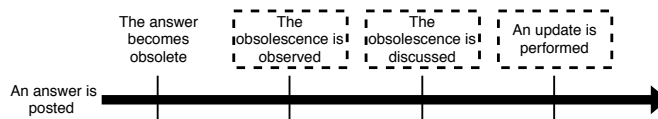


Fig. 4: One possible flow of activities that could occur after an answer becomes obsolete. Activities in dotted box are optional and might not happen in all cases.

are more likely to become obsolete. We also investigate who observes obsolete answers and what evidence do they provide to support their observations. For a better understanding of answer obsolescence on Stack Overflow, we present the possible activities that could happen after an answer becomes obsolete in Fig. 4. An answer probably becomes obsolete after some time since its creation (alternatively an answer might be obsolete even as it is being posted) (see Section 4.1). An obsolete answer probably would be observed by a user on Stack Overflow (i.e., obsolescence observation). Users may also discuss the obsolescence afterwards and update their answers correspondingly. In this study, we examine the obsolete answers that were observed by users, by them leaving a comment to indicate that the answer is obsolete. We also study whether and how users update obsolete answers.

Based on our study, obsolete answers could be categorized into two classes: *legacy* or *invalid*. We consider an obsolete answer as a *legacy* answer if it can still be used or applied, but it may not be recommended anymore since a newer answer might be better or more appropriate. For example, a comment⁶ (as shown in Fig. 5) points out that an answer is “obsolete in Rails \geq 3.0.0”, which indicates that the accepted answer only applies to Rails version 3.0.0 or below. Nevertheless, users who use earlier versions may find this answer still useful. On the other hand, an *invalid* answer indicates that the obsolete answer is not valid or it does not work anymore. Users who might have successfully applied the particular answer earlier would now run into errors or complete failures. One example of an invalid answer is related to an old http protocol (such as RFC 2616⁷), which is deprecated. For example, a comment⁸ (shown in Fig. 6) mentions that “RFC 2616 has been obsoleted”.

@Simon you are right. explicit HTML escaping with `h` is obsolete in Rails \geq 3.0.0 -- i've removed all references to `h` from my answer :) – Christoph Schiessl Dec 8 '13 at 12:40

Fig. 5: A comment indicates legacy Rails version.

a) the important point about GET is safeness, not idempotence, b) @Abhijeet: RFC 2616 has been obsoleted in 2014; see RF 7230ff. – Julian Reschke May 6 '16 at 6:16

Fig. 6: A comment indicates an invalid protocol.

Thus, we are interested in investigating obsolete answers on Stack Overflow, to understand the types of obsolescence that happen and to provide some insights into addressing the obsolescence of answers.

6. <https://stackoverflow.com/questions/186508/>

7. <https://www.ietf.org/rfc/rfc2616.txt>

8. <https://stackoverflow.com/questions/29648972/>

3 DATA COLLECTION

In this section, we describe how we collect the dataset that we used to answer our research questions.

To understand the obsolescence of answers on Stack Overflow, we need to identify answer threads (both accepted and non-accepted answers) with obsolete knowledge. As we introduce in Section 2, users occasionally leave comments to indicate that an answer is obsolete (see Fig. 1). Based on this observation, we identify answer threads that have obsolete knowledge using the following selection criteria:

- 1) A comment in an answer thread has the keywords “deprecated”, “outdated”, “obsolete” or “out of date”.
- 2) The keyword (“deprecated”, “outdated”, “obsolete” or “out of date”) does not appear in the question (including the question title and body) of its thread or any of its answers. The reason behind this criteria is that if the keyword appears in the content of a question or an answer, it may indicate that the question or answer itself is related to an “obsolete” topic rather than being a sign that the knowledge is likely obsolete.

The purpose of our selection criteria is not to collect all possible answer threads with obsolete knowledge, but to collect sufficient data for a comprehensive analysis, while minimizing the bias that is caused by false positives. The accuracy of our heuristic-based approach is 80.2% based on our manual verification from a statistically representative sampling with a 95% confidence level and a 5% confidence interval.

We downloaded the Stack Overflow data from archive.org⁹. The data was published on August 31, 2017 by the Stack Exchange community. The data contains information about badges, comments, post history, post links, posts, tags, users, and votes. Using our selection criteria, we ended up with 52,177 answer threads, which include 58,201 comments that mention obsolescence. These collected threads span 12,629 tags.

4 CASE STUDY RESULTS

4.1 RQ1: What happens when an answer is observed as obsolete?

Motivation: It is very important to keep answers up-to-date on Stack Overflow as we noted in Section 1. However, it is not known how the Stack Overflow community handles obsolete answers. In this RQ, we are interested in examining how the Stack Overflow community deals with obsolete answers after such obsolescences are observed. More specifically, we would like to investigate the activities that occur after someone observes the obsolescence of an answer. Through such analysis, we expect to provide an overview of how the community handles the obsolescence of answers once they are observed and a reasonable understanding of the severity of the answer obsolescence problem.

Approach: There are two types of actions that might occur after an answer is observed to be obsolete: 1) updating the

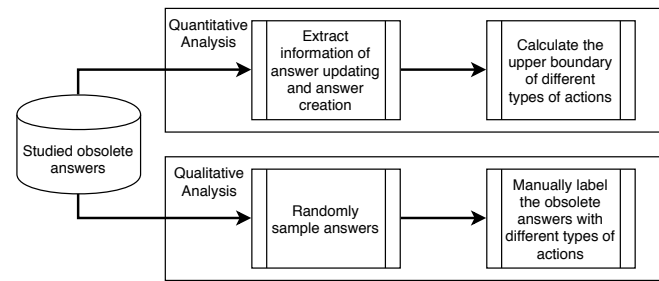


Fig. 7: The overview approach of RQ1.

obsolete answer (*update*); 2) creating a new updated answer (*new*). As a result of the above two types of actions, another action might occur, that is the switching of the accepted answer (*switch*). For example, the original asker may cancel the currently accepted answer and mark an updated one or a newly created one as the accepted answer. To understand what occurs after an obsolescence is observed, we perform both quantitative and qualitative analysis. An overview of the approach is presented in Fig. 7.

In the quantitative analysis, we captured an overall picture about when the obsolescence is observed and how users react to the obsolescence observation in terms of the three types of actions (i.e., *update*, *new*, and *switch*). To compute the number of cases in which users update the obsolete answer (type *update*), we counted the number of obsolete answers that have been edited after an obsolescence observation. Such a number gives us an upper bound estimate since updating an obsolete answer is not the only reason for editing an existing answer. We computed the number of type *new*, using a similar way as type *update*, i.e., computing an upper bound estimate. Adding updated information is one possible reason for creating a new answer, but there could be other reasons, such as adding an alternative answer. Thus, by computing the number of question threads that have new answers after an obsolescence observation, we were able to get an upper bound on the number of instances of type *new*. We were able to compute the number of type *switch* instances based on the historical records of answers. However, we did not find any case of type *switch*. Therefore we focus the rest of our analysis on type *update* and *new*.

In the qualitative analysis, we performed a manual study to calculate the exact occurrences of type *update* and *new* actions. We randomly sampled a statistically representative sample of 384 obsolete answers (including all their associated comments) from our studied 52,177 obsolete answers using a 95% confidence level with a 5% confidence interval. We manually performed a lightweight open coding-like process [5], [6] to check the sampled answers, their edit records, and the associated comments and other answers in the same question thread in order to label the types (*update* and *new*) of performed actions. We also recorded the time for users to react.

This process involves 3 phases and is performed by the first two authors (i.e., A1–A2) in this paper:

- Phase I: A1 derived a draft list of types of performed actions based on 50 random answers. Then, A1 and A2 use the draft list to categorize the answers col-

9. www.archive.org/details/stackexchange

laboratively. During this phase the types are revised and refined.

- Phase II: A1 and A2 independently applied the resulting types from Phase I to categorize all 384 answers. A1 & A2 took notes regarding the deficiency or ambiguity of the labeling for obsolete answers.
- Phase III: A1, A2 discussed the coding results that were obtained in Phase II to resolve any disagreements until a consensus was reached. No new labels were added during this discussion. The interrater agreement of this coding process has a Cohen's kappa of 0.97, which indicates that the agreement level is high [7].

4.1.1 Quantitative Analysis

More than half of the studied obsolete answers were probably already obsolete when they were first posted. Fig. 8 presents the time gap between the answer creation time and the time at which the obsolescence observation was noted. An interesting observation is that 58.4% of the studied answers were noted as obsolete within 24 hours after their creation. This suggests that more than half of the answers probably were already obsolete when they were first posted. One possible explanation is that even the answerer himself/herself does not realize that their answer is obsolete. For example, Fig. 9 shows an answerer¹⁰ who was using an obsolete API in his original answer. A commenter pointed out within 2 minutes that the answer is obsolete, then the answerer updated his/her answer.

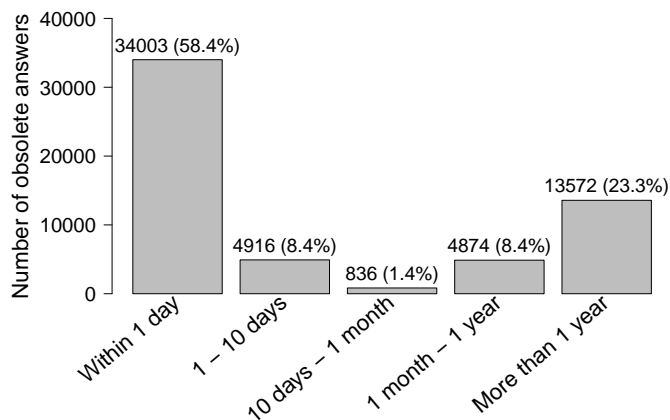


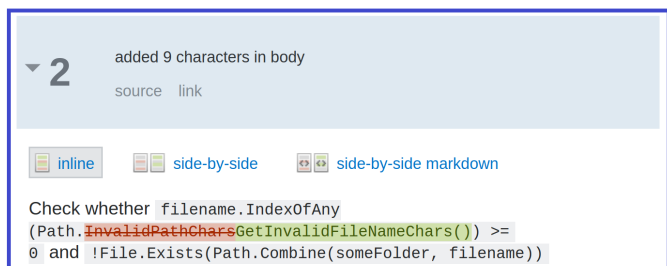
Fig. 8: Number of obsolete answers vs number of days to point out the obsolescence of the answers.

More than half of the users do not update their answers or add new answers after their answers are noted as obsolete. In terms of an upper bound estimation, 46% of the studied obsolete answers were either updated (type *update*) or added with new answers (type *new*). More specifically, less than 27.4% (upper bound) of the obsolete answers got updated after being noted as obsolete, and in 33.3% of the cases users added new answers. We also check the editing records of the accepted answers. We observe that 44.1% of the studied obsolete answers are the accepted answers. We find that 30.7% of the obsolete accepted answers got updated (type *update*) after being noted as obsolete, while

10. <https://stackoverflow.com/questions/4650483/>



Answer thread



Answer revision history

Fig. 9: An example of an answer whose poster didn't realize his answer was obsolete when he created the answer.

only 24.8% of non-accepted answers got updated. These findings suggest that accepted answers are more likely to be updated after an obsolescence was noted compared with non-accepted answers. Nevertheless, it is interesting to note that users still do read unaccepted answers and do note their obsolescence (indicating the importance of all answers not just the accepted ones). Future studies of Stack overflow should also explore non-accepted answers instead of mostly focusing on accepted answers.

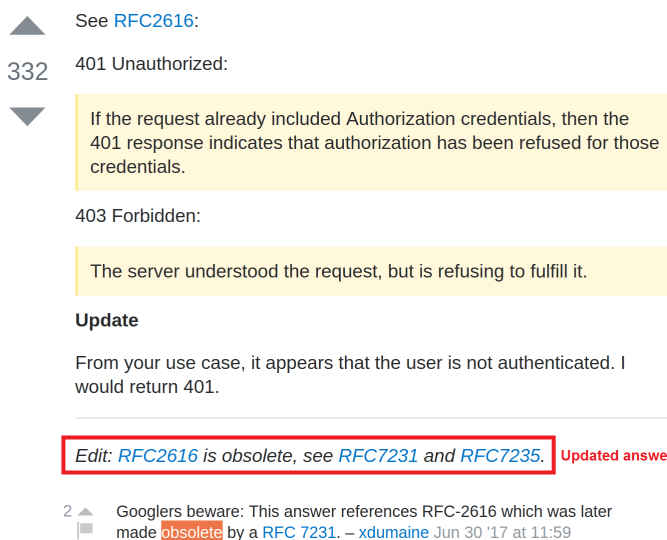


Fig. 10: An example of an obsolete answer that was updated.

4.1.2 Qualitative Analysis

Users updated their obsolete answers in 23.5% of the cases and added new answers in 4.9% of the cases. On average,

it took 101 days for users to react to an observed obsolete answer. Through a manual study, we observed that in 4.9% of the cases users added new answers, and that 23.5% of the answerers updated their obsolete answers. For example, we present a case¹¹ in Fig. 10. The answer was edited on August 11, 2017 to update the obsolete answer (i.e., information about a protocol). We also notice, on average, that it took 101 days for users to add new answers or update existing answers.

More than half of the studied obsolete answers were probably already obsolete when they were first posted. Most users did not update obsolete answers or add new answers to address the obsolescence of an answer. Even for users who performed actions to deal with the obsolete answers, on average it took them 101 days after the obsolescence of the answer was noted.

4.2 RQ2: Whether answers that are associated with particular tags are more likely to become obsolete?

Motivation: Some particular topics (i.e., associated Stack Overflow tags) evolve more rapidly than others. For example, Android is evolving at a rather rapid pace [3]. Such rapid evolution may lead to a higher likelihood of the answers of such tags to become obsolete. Therefore, in this RQ, we examine which topics (i.e., tags in our study) of answers are more prone to have obsolete answers. In addition, various reasons could lead to obsolescence (e.g., a release of a new version of a framework, such as the .NET framework, might lead to the deprecation of some APIs of the earlier versions). We are also interested in investigating why answers on Stack Overflow become obsolete. By understanding this, we could provide some suggestions for the answer seekers when they search for answers on Stack Overflow (e.g., which answers relative to their associated tags require more caution since they have a higher tendency to become obsolete).

Approach: We first conduct a quantitative analysis to examine which tags are more likely to have obsolete answers. To understand which tags are prone to have obsolete answers, we compute the number of obsolete answers to questions that are associated with a particular tag and *normalize this number by dividing with the total number of obsolete answers in our study*.

We then perform a qualitative analysis to study the reasons of answer obsolescence. In this experiment, we randomly select 384 answers (including all their associated comments) out of the 52,177 answers, in order to achieve a confidence level of 95% with a confidence interval of 5%. We manually derive and categorize the obsolescence reason from the randomly sampled answers threads. We perform a lightweight open coding-like process [5], [6] as mentioned in RQ1 to identify the reasons of obsolescence. The inter-rater agreement of this coding process has a Cohen's kappa of 0.79, which indicates that the agreement level is substantial [7].

During our manual study process, we also label whether the obsolescence is a legacy or invalid (see Section 2).

11. <https://stackoverflow.com/questions/3297081/>

TABLE 1: Types of obsolescence reasons.

Type	Definition	Example
Programming Language	Obsolescence is caused by upgraded features of the programming language.	A comment points out that the -client option is ignored by a 64-bit capable JDK since Java 6 ¹² .
Reference	References in an answer are obsolete.	A comment points out that the link to a whitepaper with detailed benchmarking for the Oracle TimesTen in-memory database is dead ¹³ .
Tool	Tool information is obsolete, such as an old version.	A comment points out that a solution is out of date for Microsoft Kinect SDK version 1.0 ¹⁴ .
Functionality	The way to implement a functionality is obsolete.	The function ftw() walks through the directory tree and calls fn() for each entry. A comment points out that ftw() is replaced by nftw() ¹⁵ .
Framework	Obsolescence is caused by an obsolete framework functionality.	A comment points out an obsolete answer for customizing the default templates in Django web application framework ¹⁶ .
Mobile OS	An answer becomes obsolete due to an obsolete mobile platform.	A comment points out the event handling syntax for Mono for Android 4.2 is out of date ¹⁷ .
Non-mobile OS	An answer becomes obsolete due to an obsolete non-mobile OS platform.	A comment points out that <code>expr</code> is obsolete for arithmetic in the Linux bash shell ¹⁸ .
API	An answer becomes obsolete due to APIs or libraries becoming obsolete.	The way to delete a project in Google APIs Console is mentioned to become obsolete in a comment ¹⁹ .
Protocol	An answer is obsolete because a protocol is upgraded.	A comment points out that the internet text messages RFC 822 was replaced by RFC 2822 ²⁰ .

Results: Answers that are related to certain tags (i.e., mobile app development and web development) are more likely to become obsolete. Fig. 11 ranks the tags in the order of the number of obsolete answers in one tag over the total number of obsolete answers under our study. We grouped the top 20 tags and obtained 4 general categories (ordered by their rankings in each category):

- 1) Mature infrastructure technology: mysql, c, sql;
- 2) Mobile development: android, ios objective-c, iphone;
- 3) Web development: javascript, php, jquery, html, css, ruby-on-rails, ajax, node.js;
- 4) Evolving infrastructure technology: java, python, c#, c++, .net.

Mature infrastructure technologies are technologies that existed already for a long time, such as c, sql and mysql. We observe that the number of questions that are associated with such tags drops after 2014 (see Fig. 12).

Tags in mobile development contain questions related to android or iOS app development (e.g., android & iphone),

12. <https://stackoverflow.com/questions/198651/>

13. <https://stackoverflow.com/questions/992288/>

14. <https://stackoverflow.com/questions/7840637/>

15. <https://stackoverflow.com/questions/6921813/>

16. <https://stackoverflow.com/questions/3544137/>

17. <https://stackoverflow.com/questions/11127047/>

18. <https://stackoverflow.com/questions/25179727/>

19. <https://stackoverflow.com/questions/25783347/>

20. <https://stackoverflow.com/questions/10877109/>

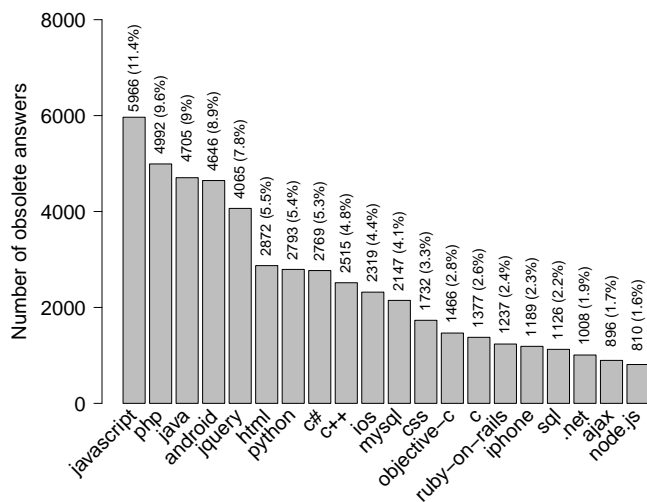


Fig. 11: The top 20 tags with the number of obsolete answers divided by the total number of obsolete answers in our study.

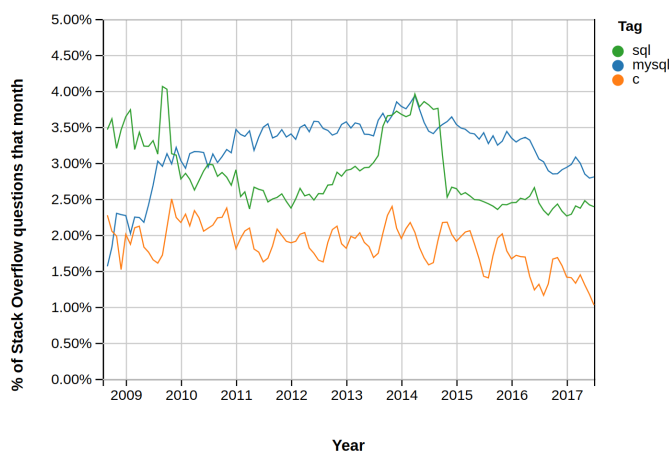


Fig. 12: The number of questions that were posted monthly in tags “c”, “sql”, and “mysql” from the creation of Stack Overflow up to August 2017.

and the tools or programming languages related to developing these apps (e.g., objective-c). Due to the popularity of mobile apps, many developers are involved in mobile app development, thus leading to an increase in the number of mobile app related questions and answers on Stack Overflow. Answers related to mobile app technologies are more likely to become obsolete because of the fast release cycles of such technologies. For instance, Google has released 15 major versions and 27 levels of API from September 2008 to Jan 2018²¹ and there are, on average, 115 API updates per month [3]. Another example is iOS. Apple has released 35 versions of iOS from Jan 2007 to Jan 2018²². Such rapid updating (in both mobile operating systems and their corresponding tools) makes the answers related to mobile development more likely to become obsolete. This phenomenon

has also been noticed by users on Meta Stack Overflow²³. For example, a user mentions that “... Android, which as a platform is only 7 years old. It has changed drastically over that time, and answers to questions that were posed 3 or 5 years ago are out of date. In some cases the answers are inappropriate or just wrong for current developers ...”²⁴. A similar situation arises to answers related to web development.

26.3% of the studied answers became obsolete due to the evolution of their associated programming language. The number of occurrence and percentage of each obsolescence reason is shown in Fig. 13, as well as the proportion of legacy or invalid obsolescence in each type. In our qualitative study, we find that most answers became obsolete due to the evolution of their associated programming language. Stack Overflow covers a broad range of questions and answers across various programming languages, and it is very common for programming languages to release new versions, thereby making the older versions possibly obsolete. For example, in a question of how to serialize and restore an unknown class in c#, an answer²⁵ suggested to use SoapFormatter instead of XmlSerializer. Another user posted a comment 3 minutes later to state that “this class is obsolete. Use BinaryFormatter instead”, including the .NET Framework version number and a reference link. Based on this finding, we recommend that users provide a version number for their answers, then Stack Overflow can note the active versions when an answer was posted and note in the UI how many versions come after it.

22.1% of the answers became obsolete due to obsolete references. Obsolete reference includes URL links, cited books, videos, and so on. Although it is convenient for a user to post an answer simply by referring to external URLs or books, it is common for references to become obsolete because the source of the reference may not be well maintained over time. This is especially a problem when users write an answer without providing too much concrete content, but instead simply offering URLs as the solution. In total, there are 5.5 million links from 7.3 million answers. To further understand the situation of obsolete reference, we randomly select 11,000 (0.2% of total URLs in Stack Overflow) URL links in answers to check if they are still available (as of April 2018). **We find that 9.5% of the selected links to be no longer accessible.** Based on this finding, Stack Overflow probably should develop mechanisms to archive a snapshot of links when they are posted.

21.1% of the studied obsolete answers are caused by outdated tools, and more than half of these outdated tools are related to IDEs. To further understand what types of tools are more likely to become obsolete, we manually study the related answer threads. Among these tools, 53.9% are related to IDEs, such as Visual Studio, Eclipse, Xcode, and Android Studio. For example, in an outdated answer²⁶ for Xcode, the commenter not only pointed out the obsolescence, but also provided an updated answer. One possible explanation is that IDEs are frequently updated in order to

21. <http://socialcompare.com/en/comparison/android-versions-comparison>

22. http://en.wikipedia.org/wiki/iOS_version_history

23. meta.stackoverflow.com

24. <https://meta.stackoverflow.com/questions/309152/>

25. <https://stackoverflow.com/questions/590722/>

26. <https://stackoverflow.com/questions/4176543/>

provide support for evolving programming languages and environments (e.g., mobile development).

Besides these types of obsolete answers, we have also seen other types of obsolescence, such as obsolete functionality, framework, operating system, API, and protocol. For example, a comment in an answer²⁷ pointed out that “iOS 5 introduced a delegate method” and therefore the existing function “scrollViewWillEndDragging:withVelocity:targetContentOffset:” is now obsolete.

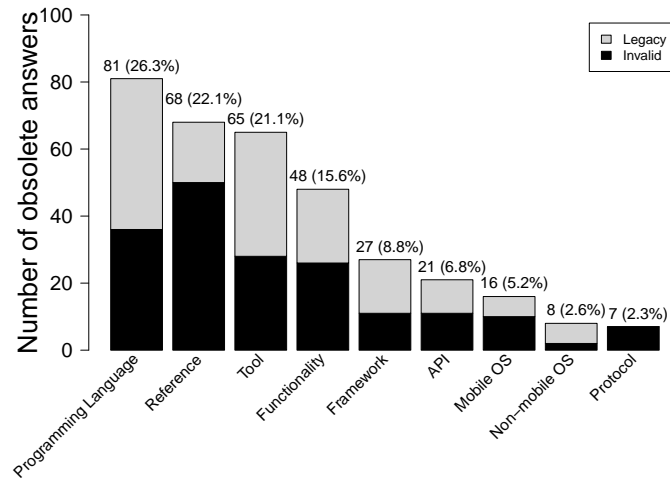


Fig. 13: Number and percentage of each obsolete type based on our manual analysis. The figure also shows the proportion of legacy (gray) and invalid (black) obsolescence.

Obsolete answers should not simply be removed as a solution because they may still be applicable to users who are using legacy technologies/systems. We find that 54.2% of the studied obsolete answers belong to the legacy category based on our study. **However, we observe that the answers that are related to protocols are all invalid in our study.** This is reasonable since once a protocol becomes obsolete, it is most likely no longer used anymore. We get the complete list of RFCs²⁸ as of May 2018. From this document, there are 8,286 RFCs, in which 1,188 RFCs are obsolete because of 1,112 newly added RFCs. We collected all answers (i.e., 21,591) containing “RFC” information from Stack Overflow, and we find that the RFCs in 10,793 answers became obsolete (being replaced by new RFCs). However, among such obsolete answers, only 611 answers updated the new RFC versions. In other words, **only 5.7% of answers mentioning obsolete RFCs have new RFC versions included in the same answer.**

Answers to questions that are associated with certain tags (e.g., web development and mobile development related tags) are more likely to become obsolete. We also find that the majority of answers become obsolete due to the evolution of their associated programming language, outdated references and tool updates. Therefore, users need to pay more attention to such answers when looking for answers on Stack Overflow.

27. <https://stackoverflow.com/questions/1677400/>

28. <https://www.ietf.org/download/rfc-index.txt>

4.3 RQ3: Who observes obsolete answers and what evidence do these observers provide?

Motivation: Uncovering obsolete knowledge on Stack Overflow is not trivial, especially if the user is not an expert in the specific knowledge domain. Therefore, it is essential to identify experts who might observe answer obsolescence and support their observations. In this RQ, we examine who identify obsolete answers. Furthermore, we are interested in investigating how they support their obsolescence observation. By analyzing these aspects, we expect to get insights into how to assist users on Stack Overflow to identify obsolete answers.

Approach: To understand who observe the obsolescence of an answer, we first perform a quantitative study on all the studied answer threads. Based on the role of the user who notes the obsolescence observation in an answer thread, we categorize observers into one of the following 5 groups:

- 1) **Asker:** the user that observe the answer obsolescence is the original asker;
- 2) **Answerer:** the user who posted the obsolete answer;
- 3) **Other answerer:** the user who posted another answer other than the obsolete one;
- 4) **Commenter:** the user who posted comments in the question thread;
- 5) **Outsider:** the user who never had any activities in the question thread.

We refer to an asker, answerer, other answerer(s), or commenter who are involved in the question thread (groups 1 – 4) as an *insider* (since they were involved earlier on in the question thread).

To understand what evidence do users provide when noting the obsolescence of an answer, we performed a qualitative study. We used the randomly selected answers from RQ2 after removing the false positives. We manually extracted and categorized the evidence of obsolescence from the sampled answers. We performed a lightweight open coding-like process [5], [6] as mentioned in RQ2. We categorized the support evidence for obsolete answers into 8 types, as shown in Table 2. The inter-rater agreement of this coding process has a Cohen’s kappa of 0.957, which indicates that the agreement level is high [7].

TABLE 2: Types of support evidence for an obsolescence observation.

Type	Definition
Provide updated Info	The user provides updated information as an explanation why an answer is obsolete.
Highlight time	The user mentions the time when the answer worked or the time interval since it worked.
Provide version Info	The user mentions the version number of either the obsolete knowledge (e.g., framework) or the updated information.
Provide link	The user posts a link as a further reference to her/his obsolescence observation.
Refer other answer	The user points to another answer on Stack Overflow to support why the current answer is obsolete.
Refer this answer	The user points to this answer because it updated the obsolete content.
Provide running error	The user shows the running error due to the obsolescence.
No support	No supportive material is given to prove the answer is obsolete. The user simply claims that something is obsolete.

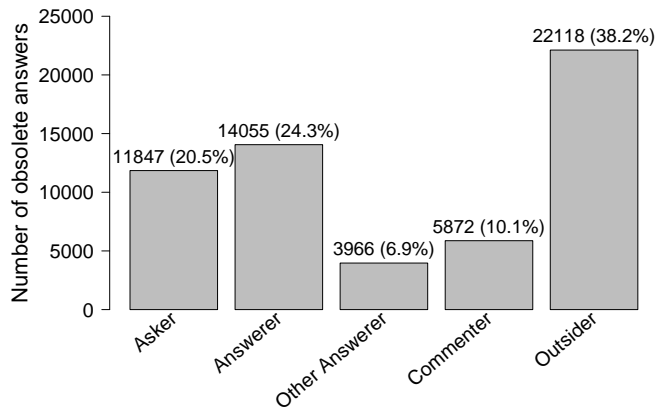


Fig. 14: The number (as well as the percentage) of the obsolete answers that are observed by each type of user. The role of users are assigned in following priority: asker > answerer > other answerer > commenter > other user. For example, if a user has multiple roles, such as an answerer and a commenter, we categorize the user as an answerer.

Results: The obsolescences of answers are more frequently observed by outsiders (38.2%), compared to askers (20.5%) and answers (24.3%) The number and proportion of obsolete answers that were observed by each group of users (i.e., asker, answerer, other answerer, commenter, and outsider) are shown in Fig. 14. Only 24.3% of the obsolete answers were observed by answerers. 10.1% of the obsolete answers were observed by commenters. 6.9% of the obsolete answers were observed by other answerers in the same question thread. 20.5% of the obsolete answers were observed by askers. The lowest proportion among the insiders are other answerers. One possible explanation of such a low proportion for other answerers is that they may never pay attention to other answers in the same question thread to check if these answers are obsolete. The rest of the obsolete answers (38.2%) are observed by users who have never participated in the discussion before observing that the answer is obsolete.

In summary, only 24.3% of the obsolete answers were observed by answerers. One possible reason is that some answerers are no longer active on Stack Overflow. Another possible reason is that even if the answerers are still active on Stack Overflow, they may not really want to maintain of their answers after a long period of time. Even worse, they may not even be active in that domain anymore. For example, one user asked how to handle obsolete answers²⁹, and one commenter mentioned that *“Two years down the line I don’t want to have to regularly rework my answers. I might not even be active in that field anymore”*. Therefore, it’s very important for Stack Overflow to encourage the whole community, not just the answerers to maintain answers by taking care of obsolete answers.

The majority (79.5%) of the obsolete observations are supported with evidence (e.g., updated information, a version information, or a reference). Fig. 15 shows the proportion of each type of supporting evidence for obsolescence observations. We observe that in the majority of cases, users

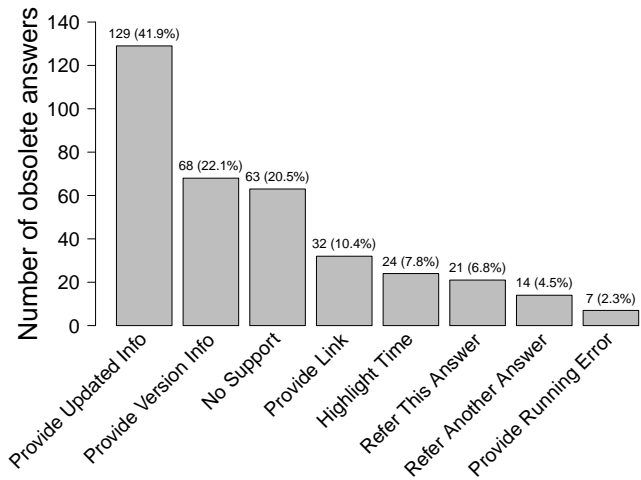


Fig. 15: The proportion of each type of evidence that users provide when pointing out obsolescence.

273

```

As of Numpy 1.9, the easiest and fastest method is to simply use
numpy.unique, which now has a return_counts keyword argument:

import numpy as np

x = np.array([1,1,1,2,2,2,5,25,1,1])
unique, counts = np.unique(x, return_counts=True)

print np.asarray((unique, counts)).T

If you are using ActivePython, then numpy is very probably out of date.
Check current numpy version with pip list AND pypm list, then
run pypm uninstall numpy and pip install numpy. -
KrisWebDev Dec 21 '15 at 9:18

```

Fig. 16: An example of a comment pointing out obsolete knowledge by providing updated information.

provide supporting evidence (e.g., updated information and a version information). For example, in 41.9% of the cases, users provide updated information about the obsolete answers. An example of a comment with updated information is shown in Fig. 16. In this comment³⁰, the user not only pointed out that numpy is out of date, but also provided the code to check the numpy version in the code to install the latest version. Such cases are not rare; we observe that 48.7% of cases a solution (an updated answer) is provided in the comments. Furthermore, version numbers are also used by some users to support obsolescence observation. Once a version number is given, it is convenient for users to identify the obsolete knowledge. We find that 22.1% of obsolescence observations have mentioned version numbers. For example, in an answer³¹ that uses AutoMapper (a convention-based object-to-object mapper and transformer for .NET), one comment started with *“as of AutoMapper 4.2 Mapper.CreateMap() is now obsolete ...”*. However, we find that 20.5% of obsolescence observations do not provide any supporting evidence. During our qualitative study, we find other types of support for obsolescence observations. For example, 7.8% of obsolescence observations are supported by highlighting time information related to the obsolescence.

30. <https://stackoverflow.com/questions/25943480/>

31. <https://stackoverflow.com/questions/6825394/>

29. <https://meta.stackexchange.com/questions/11705/>

The majority of the obsolete answers were not observed by the original answerers. To help resolve obsolete answers, Stack Overflow should develop mechanisms to encourage the whole community to maintain and flag obsolete answers. We also find that most obsolescence observations are supported by evidence.

5 HIGHLIGHTS OF THE IMPLICATIONS OF OUR FINDINGS

Stack Overflow should develop mechanisms (e.g., rewarding badges or reputation scores) to encourage users (especially question thread insiders) to pay more attention to the obsolescence of answers (their own or others') and make efforts to maintain any obsolete answers. In RQ1, we find that only around 1 out of 4 users updated their answers when their answers were noted as obsolete. Moreover, it took users over 3 months on average (i.e., 101 days) to update their answers or add new updated answers. In other words, users do not pay much attention to the obsolescence of their answers and do not frequently maintain their answers after they create them. For example, a comment of an obsolete answer³² mentioned that the answer was obsolete and asked the answerer to update it. The answerer replied in comment "Feel free to update the answer yourself, if you like. I honestly would, but I don't have the time." Thus, we suggest that Stack Overflow should develop mechanisms to encourage users to pay more attention to their posted answers and maintain their answers. For example, Stack Overflow could create certain badges which are rewarded to users who help maintain obsolete answers or reward reputation scores to users who identify and/or maintain obsolete answers. More importantly, Stack Overflow should encourage the whole community (i.e., not just the answer owners or question askers) to maintain obsolete answers due to the large number of posts on the site. Another direction is probably to develop mechanisms to identify the obsolete answers automatically. For example, Tran et al. [8] automatically detected outdated information on Wikipedia by using pattern-based fact extraction from both Wikipedia and the web. Future work could also be done to detect obsolete answers on Stack Overflow.

Stack Overflow should develop mechanisms to manage obsolete references. In RQ2, we find that a major type of answer obsolescence is outdated reference. For example, 9.5% of the studied links are not accessible any more. Wikipedia provides both dead link templates and other inline cleanup tags (such as obsolete source)³³ to manage links. This mechanism could be useful to Stack Overflow to mitigate the obsolescence problem.

Answerers are encouraged to include information of the valid version or the time of their knowledge when creating answers. In RQ3, we find that 79.5% of the obsolete answers were pointed out with supportive evidence, such as obsolete time and version information. Such information is very helpful for answer seekers to verify whether the knowledge in the answers is still valid or not.

32. <https://stackoverflow.com/questions/13088385/>

33. https://en.wikipedia.org/wiki/Template:Dead_link

Answer seekers are encouraged to carefully go through the comments that are associated with answers in case these answers become obsolete, especially for answers that are related to web and mobile development. In RQ2, we observe that answers related to some tags are more likely to become obsolete, such as tags that are related to mobile development (e.g., android and ios) and web development (e.g., php). Therefore, answer seekers are encouraged to pay more attention when searching for answers that are related to web and mobile development. One actionable way is to go through the comments under accepted answers or other answers, which may have useful information to indicate whether the answer has become obsolete or not. Even more, we observe that 48.7% of the cases provided a solution (an updated answer) in the comments. In addition, comments with high scores are highly recommended to review, since we observe that 73.5% of comments that indicate the obsolescence are the top 1 scored comments for the obsolete answers.

To understand whether our research uncovered a relevant problem on Stack Overflow and our findings are useful for Stack Overflow, we shared our findings with the Stack Overflow team. They concurred with our findings and mentioned it is interesting to see the breakdown of this problem ("obsolete info is an ongoing issue on the site, so it's interesting to see this breakdown of how that issue manifests itself."). Moreover, they are specifically interested in the analysis about the version information of platforms and programming languages. The Stack Overflow team was also interested in implementing tags to illustrate the current version of the answer. We plan to continue working with Stack Overflow to solve/alleviate the obsolete problem.

6 THREATS TO VALIDITY

External validity: Threats to external validity are related 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. To alleviate this threat, more Q&A websites should be studied in the future. We needed to conduct several qualitative analysis in our RQs; however, it is impossible to manually study all answers. To minimize the bias when conducting our qualitative analysis, we took statistically representative random samples of all relevant revisions, in order to ensure a 95% confidence level and 5% confidence interval for our observations [9].

Internal validity: Threats to internal validity are related to experimenter bias and errors. Our study involved qualitative analysis in RQs. To reduce the bias, each answer was labeled by two of the authors and discrepancies were discussed until a consensus was reached. We also showed that the level of inter-rater agreement of the qualitative studies is high (i.e., the values of Cohen's kappa ranges from 0.786 to 0.978). Another threat to our study is related to our data collection process. Due to the large number of answers and lack of mechanism on Stack Overflow to identify obsolete answers, we use a heuristic-based approach to uncover obsolete answers. The accuracy of our heuristic-based approach is 80.2% based on our manual verification, which implies that there may be noise in our quantitative

study. Hence we followed all presented quantitative studies with qualitative studies of randomly representative samples. Future study should develop a more accurate method to identify the obsolescence of an answer on Stack Overflow.

7 RELATED WORK

7.1 Understanding and Improving the Quality of Posts On Stack Overflow

One significant challenge that Q&A websites have is ensuring the quality of their knowledge [10]. Therefore, numerous studies have been done to better understand and improve the quality of knowledge on Q&A websites. Asaduzzaman et al. performed a study on 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) [11]. Yao et al. found that the quality of an answer is highly correlated with the quality of its question and developed algorithms to identify high-quality posts [12], [13]. Ponzanelli et al. studied the relationship between a set of proposed factors and the quality of a post on Stack Overflow [14], [15]. Duijn et al. found that the code/text ratio is the most important factor in determining the quality of a question on Stack Overflow [16]. Calefato et al. found that a classifier that is built only on text features (e.g., length of the body) could achieve good performance [17]. Zhang et al. conducted an empirical study on the prevalence and severity of API misuse on Stack Overflow [18]. They found that even posts that are accepted as correct answers or upvoted by other users are not necessarily more reliable than other posts in terms of API misuse. Wang et al. investigated various factors (e.g., factors related to a question, an answer and a user) that potentially affect the speed of getting an accepted answer and found that the most important factor is the activity level of the answerer community [19]. Chen et al. proposed a deep learning approach to help users on Stack Overflow fix grammar issues based on prior editing records [20].

Prior studies define the quality of content on Stack Overflow more from the presentation aspect (e.g., code and text ratio, length of text). However, prior studies never consider the quality of content in term of the time aspect; namely, obsolescence. Different from prior studies, we are the first study to investigate the characteristics and phenomenon of obsolete knowledge on Stack Overflow.

7.2 API Obsolescence in Software Engineering

Obsolescence is a common issue for software systems. Technology consulting firms estimate that 180-200 billion lines of legacy code is still in active use [21]. One reason for obsolescence is that the used APIs become obsolete due to deprecation. Khadka et al. also found that one important reason for systems to become legacy is due to the evolution of programming languages, which is compatible with our finding in RQ2 [21]. A significant amount of studies have been done on API deprecation. Zhou et al. proposed a lightweight version-sensitive framework to detect deprecated API usages in source code examples on the Stack Overflow [22]. Robbes et al. studied how developers react

to API deprecation in Smalltalk ecosystem [23]. They found that a number of API changes caused by deprecation can have a very large impact on the ecosystem (i.e., in terms of the projects or developers that are impacted by the change). McDonnell et al. studied how APIs evolved in the Android ecosystem and found that 28% of API calls are outdated with a 16 months lag time (i.e., the time between commit and the API release) [3]. 22% of the outdated API usages eventually upgrade to use newer API versions, but the propagation time (i.e., the time between the API release and the client adaptation) is about 14 months.

Different from prior studies, which only focused on APIs, we focus on the obsolescence of all answers on Stack Overflow. We also investigate the characteristics of such obsolete answers.

7.3 Leveraging the Knowledge from Stack Overflow

Stack Overflow accumulates a large amount of knowledge and researchers have done a remarkable number of studies to leverage the knowledge on Stack Overflow to facilitate development and maintenance activities. Zagalsky et al. built a code recommendation tool to recommend high-quality code [24]. Abdalkareem et al. studied the characteristics of source code reuse on Stack Overflow [25]. Treude et al. presented an approach to automatically enrich API documentation with “insight sentences” extracted from Stack Overflow [26]. Vassallo et al. extracted discussions from Stack Overflow and used the extracted data to generate JavaDoc automatically [27]. Wong et al. leveraged questions and answers on Stack Overflow to automatically generate comments in system source code [28]. Gao et al. proposed an automated approach to fix recurring crash bugs by leveraging information (e.g., questions with similar crash traces) on Q&A websites [29]. Wang et al. leveraged the tag information in Stack Overflow to infer semantically related software terms [30]. Wang et al. leveraged the textual information of a question that a user poses to help users infer associated tags for the posted question [30].

Instead of leveraging the knowledge from Stack Overflow, we study the knowledge obsolescence on Stack Overflow. Our finding indicates that many answers on Stack Overflow may become obsolete, which may affect the quality of the content that is produced by the above-mentioned techniques. Therefore, further research should take caution when leveraging the knowledge from Stack Overflow.

8 CONCLUSION

In this paper, we present an empirical study of the obsolete knowledge on Stack Overflow, as the first step towards understanding the evolution of knowledge on Stack Overflow. We find that: 1) Answers in certain tags (e.g., web and mobile development tags) are more likely to become obsolete mainly due to the evolution of their associated programming language, and outdated references and tool updates. 2) Most of the studied obsolete answers are pointed out by non-answerers and are supported by evidence. 3) When an obsolete answer is identified, only a small proportion of such answers are updated afterwards. More importantly, more than half of the obsolete answers were

probably already obsolete when they were posted. Based on our findings, we offer the following suggestions: 1) Stack Overflow should develop mechanisms (i.e., incentive systems) to encourage the whole community to identify and/or maintain obsolete answers. 2) Answers are encouraged to include information of the valid version or time of the knowledge when creating answers. 3) Answer seekers are encouraged to go through all the information in an answer thread carefully in case these answers become obsolete, especially for the answers that are related to web and mobile development.

There are two possible directions for future work. First, we encourage future studies to develop advanced approaches to detect obsolete knowledge on Stack Overflow. For example, machine learning techniques (e.g., Word2vec [31]) can be employed to detect the comments that indicate obsolescence based on the semantic meaning of the text instead of keywords matching. Second, we encourage future studies to develop approaches to extract useful information from the comments so that answer seekers could find the useful information from the list of long and unorganized comments easily.

REFERENCES

- [1] C. Sadowski, K. T. Stolee, and S. Elbaum, "How developers search for code: A case study," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. FSE '15, 2015.
- [2] Seralo, "Android versions comparison," <http://socialcompare.com/en/comparison/android-versions-comparison>, 2018.
- [3] T. McDonnell, B. Ray, and M. Kim, "An empirical study of api stability and adoption in the android ecosystem," in *Proceedings of the 2013 IEEE International Conference on Software Maintenance*, ser. ICSM '13, 2013, pp. 70–79.
- [4] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, "How do developers utilize source code from stack overflow?" *Empirical Software Engineering*, 2018.
- [5] 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.
- [6] 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.
- [7] A. J. Viera, J. M. Garrett *et al.*, "Understanding interobserver agreement: the kappa statistic," *Fam Med*, vol. 37, no. 5, pp. 360–363, 2005.
- [8] T. Tran and T. H. Cao, "Automatic detection of outdated information in wikipedia infoboxes." *Research in Computing Science*, vol. 70, pp. 211–222, 2013.
- [9] S. Boslaugh and P. Watters, *Statistics in a Nutshell: A Desktop Quick Reference*, ser. In a Nutshell (O'Reilly). O'Reilly Media, 2008.
- [10] 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.
- [11] 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.
- [12] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Want a good answer? ask a good question first!" *arXiv preprint arXiv:1311.6876*, 2013.
- [13] —, "Detecting high-quality posts in community question answering sites," *Inf. Sci.*, vol. 302, pp. 70–82, 2015.
- [14] 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. QSIQ '14, 2014, pp. 343–352.
- [15] L. Ponzanelli, A. Mocci, A. Bacchelli, M. Lanza, and D. Fullerton, "Improving low quality stack overflow post detection," in *Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference on*. IEEE, 2014, pp. 541–544.
- [16] 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.
- [17] F. Calefato, F. Lanubile, and N. Novielli, "Moving to stack overflow: Best-answer prediction in legacy developer forums," in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '16, 2016, pp. 13:1–13:10.
- [18] T. Zhang, G. Upadhyaya, A. Reinhardt, H. Rajan, and M. Kim, "Are code examples on an online qa forum reliable? a study of api misuse on stack overflow," in *Proceedings of the 40th International Conference on Software Engineering*. ACM, 2018.
- [19] S. Wang, T.-H. Chen, and A. E. Hassan, "Understanding the factors for fast answers in technical q&a websites," *Empirical Software Engineering*, 2017.
- [20] C. Chen, Z. Xing, and Y. Liu, "By the community & for the community: A deep learning approach to assist collaborative editing in q&a sites," in *Proceedings of the 21st ACM Conference on Computer-Supported Cooperative Work and Social Computing*. ACM, 2018, pp. 32:1–32:21.
- [21] R. Khadka, B. V. Batlajery, A. M. Saeidi, S. Jansen, and J. Hage, "How do professionals perceive legacy systems and software modernization?" in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE '14, 2014, pp. 36–47.
- [22] J. Zhou and R. J. Walker, "API deprecation: A retrospective analysis and detection method for code examples on the web," in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE '16, 2016, pp. 266–277.
- [23] R. Robbes, M. Lungu, and D. Röthlisberger, "How do developers react to api deprecation?: The case of a smalltalk ecosystem," in *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, ser. FSE '12, 2012.
- [24] A. Zagalsky, O. Barzilay, and A. Yehudai, "Example overflow: Using social media for code recommendation," in *Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering*, ser. RSSE '12, 2012, pp. 38–42.
- [25] R. Abdalkareem, E. Shihab, and J. Rilling, "On code reuse from stackoverflow: An exploratory study on android apps," *Information and Software Technology*, vol. 88, no. Supplement C, pp. 148 – 158, 2017.
- [26] C. Treude and M. P. Robillard, "Augmenting api documentation with insights from stack overflow," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE '16, 2016, pp. 392–403.
- [27] C. Vassallo, S. Panichella, M. Di Penta, and G. Canfora, "Codes: Mining source code descriptions from developers discussions," in *Proceedings of the 22Nd International Conference on Program Comprehension*, ser. ICPC '14, 2014, pp. 106–109.
- [28] E. Wong, J. Yang, and L. Tan, "Autocomment: Mining question and answer sites for automatic comment generation," in *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '13, 2013, pp. 562–567.
- [29] Q. Gao, H. Zhang, J. Wang, Y. Xiong, L. Zhang, and H. Mei, "Fixing recurring crash bugs via analyzing Q&A sites (t)," in *Proceedings of the 2015 30th International Conference on Automated Software Engineering*, ser. ASE '15, 2015, pp. 307–318.
- [30] S. Wang, D. Lo, and L. Jiang, "Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging," in *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*. IEEE, 2012, pp. 604–607.
- [31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.