An Empirical Study of Obsolete Answers on Stack Overflow

Shreykumar Patel
Lakehead University
Thunder Bay, Canada
spatel88@lakeheadu.ca

Vardhil Patel
Lakehead University
Thunder Bay, Canada
vpatel62@lakeheadu.ca

Rutvij Kanani
Lakehead University
Thunder Bay, Canada
kananir@lakeheadu.ca

Abstract

Stack Overflow is a programming-related question and answer service for both professionals and hobbyists. It contains questions and answers about a variety of computer programming topics. Stack Overflow has amassed a massive amount of software engineering expertise. Certain knowledge in replies, on the other hand, may become obsolete as time goes on. If such obsolete answers aren't explicitly recognized and acknowledged, they can lead to confusion and unanticipated difficulties. By performing the task, we identify the characteristics of obsolete answers along with how answers become outdated as knowledge gets obsolete. As result, we identify that particularly half of the out-of-date answers (51.96%) were certainly out-of-date when they were originally submitted. Moreover, tags like node.js, swift, and android are more likely to have obsolete answers to their questions on the post.

Keywords: Stack Overflow, obsolete answers, QA website, share information

ACM Reference Format:
Shreykumar Patel, Vardhil Patel, and Rutvij Kanani. 2018. An Empirical Study of Obsolete Answers on Stack Overflow. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 Introduction

Users contribute a large quantity of knowledge to technical questions and answers (QA) websites. Developers are particularly active on these QA websites, as they are continuously confronted with new development issues that necessitate the assistance of other specialists. When people are faced with insoluble difficulties, they frequently use search engines (e.g., Google, Yahoo, Bing) to look for solutions. In answer to people' questions, search engines frequently send them to technical QA websites. One of the most popular QA websites is StackOverflow, which contains 22 million questions, 32 million answers, 82 million comments, and 62,000 tags till 22 August 2021 [14]. Stack Overflow enables programmers to pose queries and provide solutions to programming issues. Due to that, the number of questions and answers rises every single day.

Users have the opportunity to answer the question on the posted questions. But somehow, many of the answers from users are deprecated. Now a question arises, what are the deprecated answers? The term deprecated refers to a software functionality that has been superseded and therefore should be ignored. Using obsolete APIs can expose you to data breaches and slow down your system. As a result, calls to obsolete APIs should be updated promptly with calls to equivalent APIs [1]. Usually programmers use API docs given by support API libraries, as well as stackoverflow's QA website, to resolve obsolete APIs. So updating or adding new answers are required to remove deprecated answers.

When the developer is working on the project and having difficulties on task. At that moment, they go through on stack overflow by posting queries. Answerers comment on the post by suggesting solutions. While using this suggested code in their project is deprecated or outdated, but developers do not have any idea about this issue. At the same time answerers also unaware of these issues. Which causes several issues in the project like deprecated API framework or security issues. After using outdated code in the project it is difficult to maintain for developers. As a result, information on how to analyze or mitigate this condition is required.

In this study, we get 91,572 threads from the posts.xml and comments.xml using big-query and then we converted into a .csv file. After observing obsolete threads we performed qualitative analysis and quantitative analysis. The purpose of applying both analyses is to examine the results with a different quantity of datasets to see how the pattern of obsolescence replies differs between the two approaches. Here, four research questions were performed to identify the deprecated answers:

- **RQ1: What happens when an answer is observed as obsolete?** [2]
  Users upload the answers on the post while a question

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

is posted. But as the study shows, half of them posted deprecated answers to the question. Now answerers have the choice to identify the out-of-date answers posted by them, and get a chance to update the answers or add new answers. In most cases, it took an average of 122 days on average to add new answers after an answer was observed as obsolete.

- **RQ2: Are answers to questions with particular tags more likely to become obsolete than other answers?**[2]
  In a research study we observed, posts associate with tags like node.js, swift, and android are more likely to have obsolete answers to their questions on the post.
- **RQ3: What are the potential reasons for answers to become obsolete?** [2]
  There are several reasons behind the answers becoming deprecated due to various reasons like third-party libraries, corresponding programming languages, mobile operating systems, and other resources. So identifying the deprecated answers from this category gives also another option for identifying outdated answers.
- **RQ4: Who observes the obsolete answers and what evidence do they provide?** [2]
  In many cases, obsolete answers are not observed by the user who uploaded answers.

In the following study of section 3 and 4 we will discuss all research questions deeply with the dataset which we used. In each research question we will discuss the motivation, approach to solve RQ and result.

2 Background

In this section we will discuss about our understanding about stack overflow question answer mechanism and obsolete answer. To gather knowledge about stack overflow, first we went through its database description available by stack exchange community [18] and try to understand database of stack overflow and how it stores data in the background. We read description of data base on stack exchange community [18] and then went through all the tables namely Posts, Users, Comments, Badges, PostHistory, PostLinks, and PostTags. Moreover, we also understand how tables are associate to each other by looking at its class diagram which is also provided by stack exchange community [18]. However, understanding the stack overflow dataset is our key aspect to perform this task. To get the dataset we used bigquery and get the data if comments contain one of the following key word ('obsolete','deprecated','out of date','outdated'). Once dataset is ready than our next approach was to start performing research question of an 'An Empirical Study of Obsolete Answers in Stack Overflow' [2]. However, we understand we get to know that which answers are accepted and which answers are not with obsolete comments. So, we came to conclusion that so there are some obsolete answers are accepted by asker without performing it. Where as some answers are not obsolete when they were posted but due to version change or any other reason it became obsolete. There are also possibilities that outsider observe obsolete without considering answer result. When we observe comments we also get to know that, some comments are invalid.

Using quantitative dataset, first we analyze that how many days it takes to observe an answer become obsolete by users. However, whether that user is asker, answerer or any other user. By following 'An Empirical Study of Obsolete Answers in Stack Overflow' [18] we discover that how tag associate with any answer make that answer become obsolete. In our findings we discover that which tags are more likely to make an answer obsolete. Apart from it, by randomly analyzing comments we came to know the potential reasons behind obsolete answers like third party library, programming language, mobile or non mobile os, protocol, reference link or tool. Overall, our effort was to find potential reasons of an answer become obsolete. Moreover, we also want to understand reasons of obsolescence on stack overflow answers so which help to reduce threat on stack overflow due to obsolete answer.

3 Data preparation

In this section, we describe how we get the dataset to perform all 4 research questions.

3.1 Quantitative Dataset

To study the obsolescence answer on stack overflow we understand the threads which contain questions, answers and comments. Another thing is that, whose accepted id is present means their answer is accepted and whose not their accepted id is not present. We notice some comments from users who describe the answer as obsolete. Apart from it, we also notice some comments of users to get more information about obsolete answers and in Figure 1 depicts five random comments of users who have observed obsolete answers.

Approach to create a final quantitative database: First, we have downloaded the comments.xml and posts.xml from the stack overflow [16]. The data was published on June 30 2021 by the stack exchange community. Our biggest challenge is to parse such a long xml file. To parse long xml files to get obsolete information we used google colab and parsed the file line by line and collected those rows which contain comments with one of the keywords like 'outdated', 'deprecated', 'obsolete' or 'out of date'. The reason behind the criteria is that if one of the key words is present in the comments then it is more likely to become the obsolete answer.

After that we need those comments question and their answer related information. For that, We have created the

```
FileUtils.readFileToString is now deprecated
with Java 8. It is no longer recommended to
use this way.
```

```
Crispy doesn't do table layouts, and neither should you.
HTML table layouts have been deprecated for at least a decade now.
```

```
as of Rails 4, the dynamically created versions
of find_or_create_by_XXX seem to have been
deprecated in favor of find_or_create_by(),
passing in the name and value.
```

```
In the else clause, uniqueIdentifier is deprecated in iOS5, hence
use something along the line of a random UUID instead. code:
CFStringRef cfUuid = CFUUIDCreateString(NULL, CFUUIDCreate(NULL));
```

```
I added new version of getting photos,
since AssetsLibrary was deprecated.
```

Figure 1. Random obsolete comments

final dataset using google bigquery as in figure 2 demonstrated the query and then we have stored them into google drive. This sql query merges the data of 3 different tables post_question, post_answer and comments.

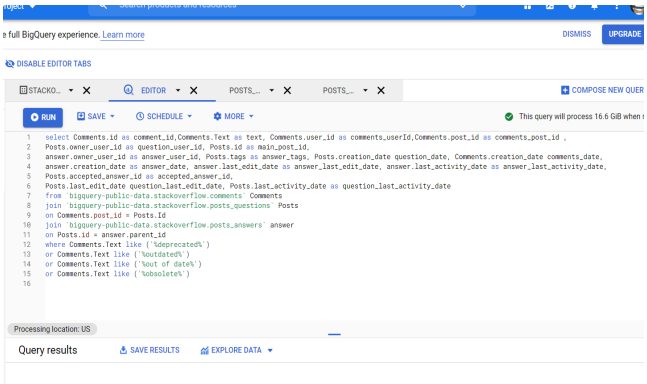

Figure 2. Big Query

We have collected the sufficient rows to understand the reason behind the obsolete answer. We have also observed bias from the comments and we considered them as false positives. To create a row which contains comments information like post's question, post's answer and post's comments we considered it as a thread. We ended up with 91572 threads with over all 103323 obsolete comments. We also collected 17340 tags associated with the threads.

3.2 Qualitative Dataset:

In this project, we also used qualitative approach for getting the good accuracy in the result. For this approach we don't need whole dataset instead we just selected certain number of rows from huge dataset. Here, we randomly selected 650 answer rows including associate comments from the "comments.csv" dataset. In addition, we also add two more columns into that new generated qualitative dataset that are "ObsoleteType" and "LegacyOrInvalid". This both columns are very crucial to determine the result so for building this both columns we added data manually by reading the comments. In the ObsoleteType column we categorise comments into seven types and in LegacyOrInvalid column we bifurcate comments into 2 types which we described in other section.

4 Research Questions

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

Motivation: The motivation behind this research question is to maintain answers on the post up-to-date as early as possible. In the given RQ, we are going to observe how much time the user will take to identify whether the answers are obsolete or not. We hope to gain knowledge of how the users handle answer depreciation once it is noticed, as well as a reasonable estimation of the seriousness of the answer obsolescence problem for Stack Overflow engineers to consider.

Approach: To handle the obsolescence in the answers users can do two things. First of all, users can edit the answer while they know that the comment is deprecated. Another option is the user can add a newly updated answer. The original asker, for example, might remove the already acceptable answer and replace it with an updated or newly constructed one. To identify what happens after the obsolescence is observed in the answer. We performed quantitative analysis and qualitative analysis.

In the quantitative analysis, we go through the whole dataset. We analyze where the obsolete answer occurred and how a user will react to that obsolete content. To handle the obsolescence user can perform updates and new comment tasks to remove the obsolescence. To identify the time which users take to remove obsolescence we took the upper bound of time and lower bound of time. Where users edited their comment by updating the method and removing obsolescence. Another way to remove deprecated answers is newly added answers. We do a similar task in the method where

we find the upper bound of the time limit. The upper bound of the time limit is considered as when the user updates or adds new comments on the post. However, a lower bound time limit is considered as the time when the user posted a comment on the post for the very first time.

Another analysis approach is qualitative analysis. In this approach, we manually observed the occurrence type of updated or newly added comments. We have taken 650 random obsolete answers out of 91572 obsolete answers.

4.1.1 Quantitative Analysis . In the quantitative analysis, we performed a task on 91572 threads from the posts.xml and comments.xml file. Where we identify how much time all this comment's users will take to update or add a new comment to remove the obsolescence from the answers. Result shows that almost half of the old replies studied were likely certainly outdated when they were submitted. So, 51.96% of answers were updated on the first day when they were posted by users.

Although, the remaining half of answers were observed as obsolete because users did not perform any task of updating or adding new comments. The ratio for obsolete content after the 1st day is still 49.04%. Another thing we discovered was that it took 122 days on average to add new answers after an answer observed as obsolete.

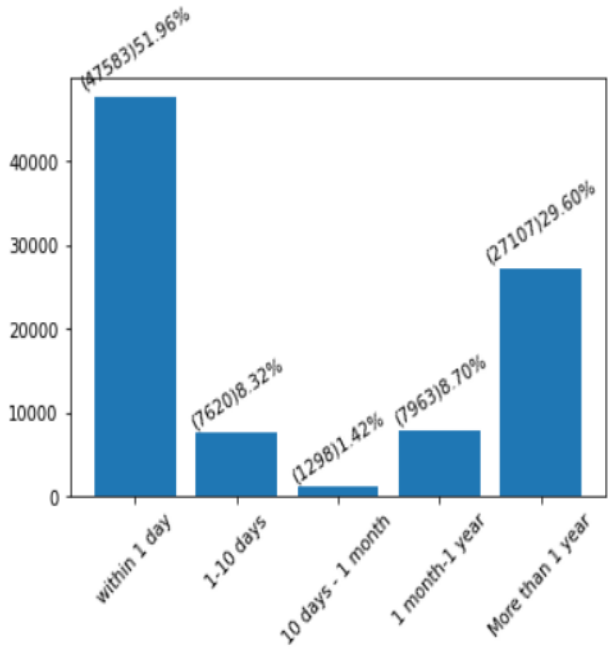

Figure 3. Number of obsolete answer observed within time

After such an obsolete answer is posted then it takes time to be observed as obsolete. As we can see in the figure 3 there are more than 50% of answers observed as obsolete within one day after it was posted and nearly 30% of answers observed as obsolete after 1 year. The answers which were observed as obsolete after 1 year majority of them are not obsolete when it was posted but due to version change either in programming language or API it became obsolete.

4.1.2 Qualitative Analysis. To perform qualitative analysis: observe updated and new added answers from user comments. By applying this approach we can identify how many comments were updated or newly added.

4.2 RQ2: Are answers to questions with particular tags more likely to become obsolete than other answers?

Motivation: Some topics (and their related Stack Overflow tags) evolve at a faster rate than others. According to McDonnell et al. Android evolves at a rate of 115 API updates every month on average [3]. So, android, for example, is evolving at a breakneck speed. As a result of this quick evolution, it's possible that the responses to such tags will become obsolete sooner rather than later. In this RQ, we survey obsolete answers associated with tags. By observing this, we were able to identify which tags contained the most obsolete answers. Like many users search their question answers by their relevant tags. So, finding the obsolete content from tags also helps us to observe the pattern of obsolescence in the answers.

Approach: We performed quantitative analysis in this RQ to perform the task. In this RQ we analyze which tags are more likely to contain answers that are no longer valid? We compute the number of obsolete responses to queries connected with a specific tag to determine which tags are prone to having obsolete replies. After that, we divided that particular tag count by the total number of answers for the same tag on the stack overflow. As a result, we observe that tags like node.js, swift, and android are more likely to have obsolete answers to their questions on the post. However, in our findings we also get 17340 unique tags from overall 313591 tags which associate with 91572 posts.

Figure4 shows the ratio of obsolete answers with a particular tag to the total number of answers with the same tag. Like node.js tag contains 0.0163% of answers observed as obsolete. Similarly 0.0142%, 0.0107%, 0.0096%, and 0.0091% of the comments of tags like swift, android, python, C respectively. In this analysis we observe the top 20 tags which contain the most obsolete answers in their comments. Meanwhile, there are other tags except for these tags also willing to contain obsolete answers in their comments.

There are several reasons behind searching questions by tags, so this approach gives results for observing the obsolete answers from particular tags. Mobile application development is very popular in the market, due to the fact that many developers are working on it currently. While developing mobile applications, developers face many challenges

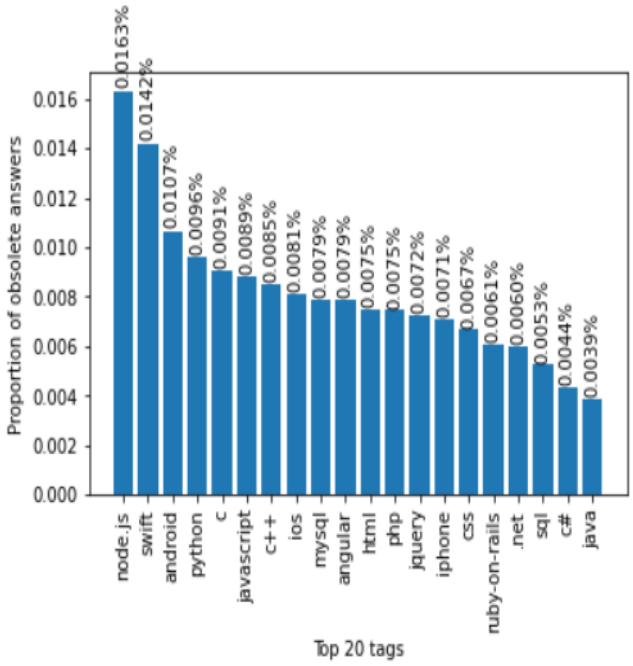

Figure 4. The top twenty tags, ranked by the percentage of answers that are outdated compared to the total number of answers in each tag.

and they refer to Stack Overflow kind of QA websites. At that moment they received answers from commentators and that most likely to become obsolete as per this research. Because we already discussed from September 2008 until August 2018, Android launched 16 major versions and 28 API levels [15] [3]. Because mobile operating systems and their accompanying technology are updated so frequently, mobile development answers are more prone to become obsolete.

4.2.1 Our Approach: We first calculate the number of tags associate with the obsolete answer. So we get total 17340 unique tags which associate with the obsolete. Then we find the top 20 tags according to their frequency. Means the tag which has more frequency we put it at the first place and then put the next tag in descending order and using this approach we also plot the bar graph as we can see in the figure 4. But to get the proportion of the tag we came to the conclusion that we divide the frequency of tag by number of time is occurs in entire stack overflow. We got the number of occurrence from entire stack overflow by using sql query on google cloud bigquery platform. To get top 20 tags with according to frequency in descending order we used nltk's FreqDist class. With the help of this class we used FreqDist's most_common method to achieve the results.

Description of figure 4 results: We used these four queries as shown in the figure 5 and sum their results. This process we have done to get the results of single tag. The

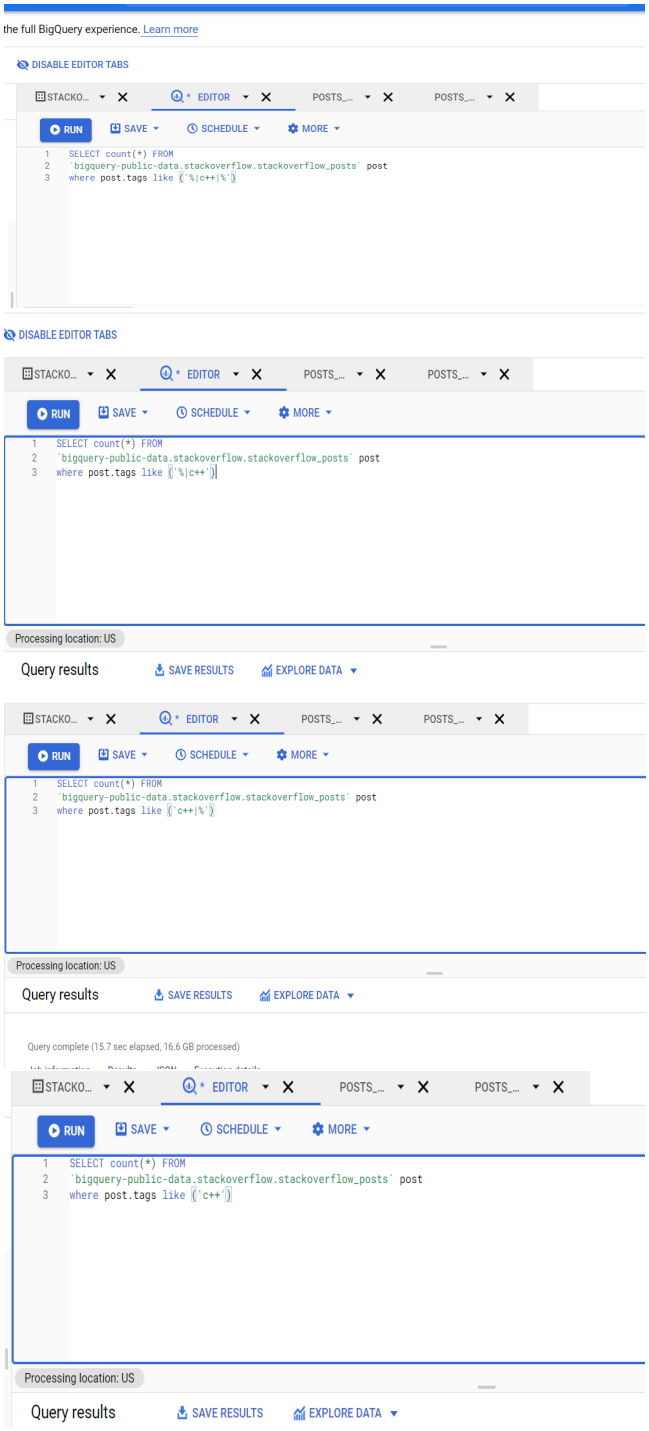

Figure 5. Count the particular tag occur in stack overflow

reason behind to use these four different queries because in the database tag come up like in figure 6 either in between the pipeline or ending with pipeline or starting with pipeline, or alone. Let us explain little more about the queries let us consider php tag if php tag is in between pipeline like '|php|' for that we used first query from figure 5, if it is appear last

in the column like '|php' for that we used 2nd query from figure 5, if it appear first 'php|' in the table row for that we used first query and if the php tag is alone like 'php' inside the row than we used last query from figure 5. After that we sum all the results for single tag. Using these approach we found the how many times particular tag appear in the entire stack overflow dataset. Similarly, we perform it with all the remaining 19 tags. After getting all 20 different tags results then we used it for dividing from obsolete comments frequency.

```
0                                    perl|sh|backticks
1                                    perl|sh|backticks
2                                    jquery|datatables
3                                               heroku
4                                  ruby-on-rails-3|rake
5                                jsf|dynamic|jsf-2|action
6                             python|datetime|int|python-2.6
7                                heroku|phantomjs|buildpack
8   sql|ruby-on-rails|ruby-on-rails-3|activerecord...
9                              php|datetime|rfc2822|rfc1123
```

Figure 6. Tags appear in database

4.3 RQ3: What are the potential reasons for answers to become obsolete?

Motivation: In this constantly changing world, there are so many factors behind obsolete answers for example version up-gradation of framework and many other factors. In this RQ we are enthusiastic to find out how these Stack Overflow answers become obsolete because if we find this, then we can easily discard those Stack Overflow obsolete answers and help learners or users to give accurate answers.

Approach: In this task we perform qualitative approach instead of quantitative approach because when we use quantitative approach, we use large amounts of data so during that time we know that, we will not get the accurate result compared to the qualitative approach so for getting good accuracy we used qualitative approach. In this task, first of all we randomly generated 650 answer rows including associate comments from a huge number of datasets. Then we add two more columns which are "ObsoleteType" and "LegacyOrInvalid" In both the columns we added the data manually by checking the comments. Here in the ObsoleteType column we were categorizing comments into seven types which are Third Party Library, Programming Language, Reference, Tool, Mobile OS, Non-mobile OS, Protocol which we described in the table below. In addition, in the LegacyOrInvalid column we categorised comments into two types: first one is Legacy and another one is Invalid. In Legacy, we describe those types of comments which the user can use as a reference even if it became obsolete while in Invalid, we keep those comments which the user cannot use as a reference when it becomes obsolete

Reason	Definition
Third Party Library	Answers become obsolete because frameworks, APIs and Third-Party Libraries become obsolete
Programming Language	Answers obsolete because features of the languages are obsolete or new versions of it.
Reference	Due to obsolete reference
Tools	Information about the tool is obsolete, for example old versions
Mobile OS	Due to obsolete mobile Operation system
Non-mobile OS	Answer obsolete due to obsolete non-mobile OS.
Protocol	Answer obsolete because of old protocol.

Table 1. Reason for obsolete answer

Here in the figure 7 diagram, we can see that in legacy, the highest chance to become obsolete is due to releasing a new version of a programming language. According to this diagram the ratio of this reason is 37.6% while for invalid the major chances to become obsolete because of utilized obsolete third-party libraries. This both reasons are so common for obsolete answers because we know that in certain time period authority releases the new version of the programming language as well as libraries so due to that upgradation sometime, they remove the certain functionality because of less usage or another way to utilise it during that time if any answerer gave the answer about that and in future if learner

cannot able to utilise it due to new version so at that time answer become obsolete. 34.5% of the answers are obsolete because of third Party library in legacy.

At the third stage we can find that the reason behind the obsolete answers is due to the evolution of mobile OS as well as tools because according to the study we find out that the percentage for both reasons are almost same which is approximately 9.5%. When we talk about the tool so that is basically talking bout the IDEs because we all know that organizations who built the IDEs, they always improve their tools by applying new features and discarding the unwanted features during this time they release the new version IDEs with the new features so due to that point of view the answers became the obsolete.

Obsolete references are also the effective reason behind the obsolete answer because we all know that in answer, answerers always give the links of other articles for referring to certain topics but due to some reasons like not maintaining the website which they refer or removing that website from the server.

According to our study, the least affected reason behind the obsolete answer is protocol because in the graph we can see that the ratio of protocol reason is very less (almost 0%).

This graph is very according to the selected rows because in this we selected 650 rows and that all are random. next time, if we select rows randomly, then the graph might be different than this so in qualitative approach the result always depends on the data(selected rows).

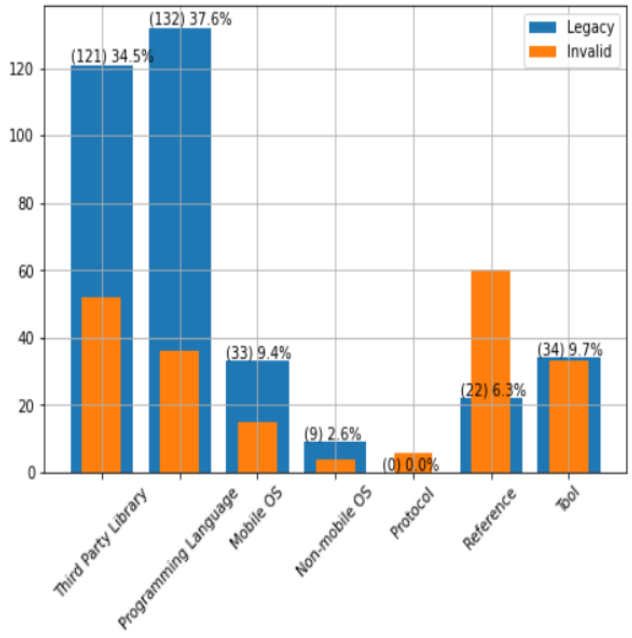

Figure 7. Reason behind obsolescence in answers in terms of percentage with total number. The fraction of legacy (blue) and invalid (orange) obsolescence is also shown in the graph.

4.4 RQ4: Who observes obsolete answers and what evidence do they provide?

Motivation: Q&A websites are very useful for those users who want to learn new things or they can find the answers of their questions through this website and Stack Overflow is the best website for learning new things but sometime, due to obsolete answers learners have to face certain problems and they cannot able to find out the desire answers from it. During that time they need experts who can help them to find out the obsolete answers as well as guide them. So in this RQ we investigate the observers who can give the support by observing the answers.

Approach: To understand who observes the obsolescence of an answer, we perform quantitative analysis. We categorize them into three different groups: asker, answerer and all other commenters.

1) Asker : The user who posted the question and observed the answer as obsolete.

2) Answerer : The user who gives the answer of a question and observes their own answer as obsolete.

3) All Other Commenter: This group consists of all the commenter like an outsider, other than asker or answerer.

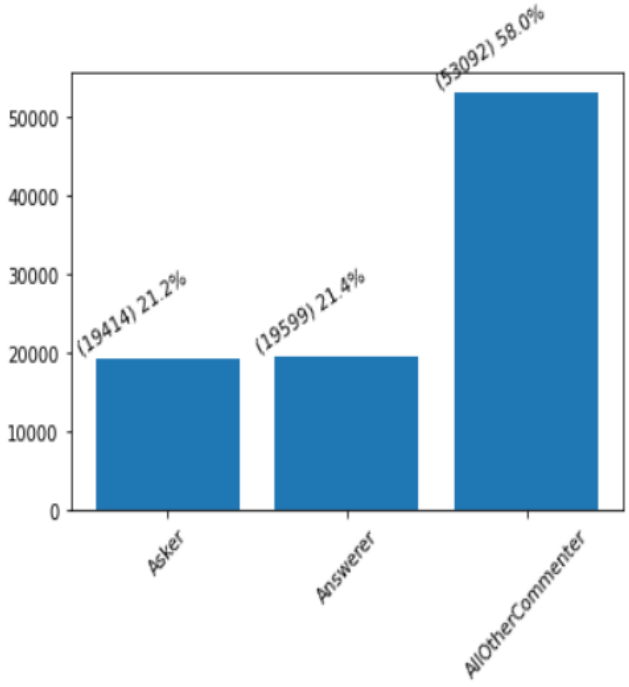

Figure 8. The total number of obsolete answers noticed by each user type (including the percentage).

Here in figure 8 we can see that, around 42% observers are those who involved in the question where as 58% users

are either outsider or other commenter.

Description of figure 8 results: First we understand who is asker and who is answerer. Than we came to know that if question user id and comment used id is the same than we consider them as asker. Where as answerer user id and comment user is the same then we consider them as answerer. While rest of other commentors we put in the third group.

5 Flow of our analysis

To perform analysis on obsolete answers in stack overflow we observed the dataset and designed this flowchart to perform the task.

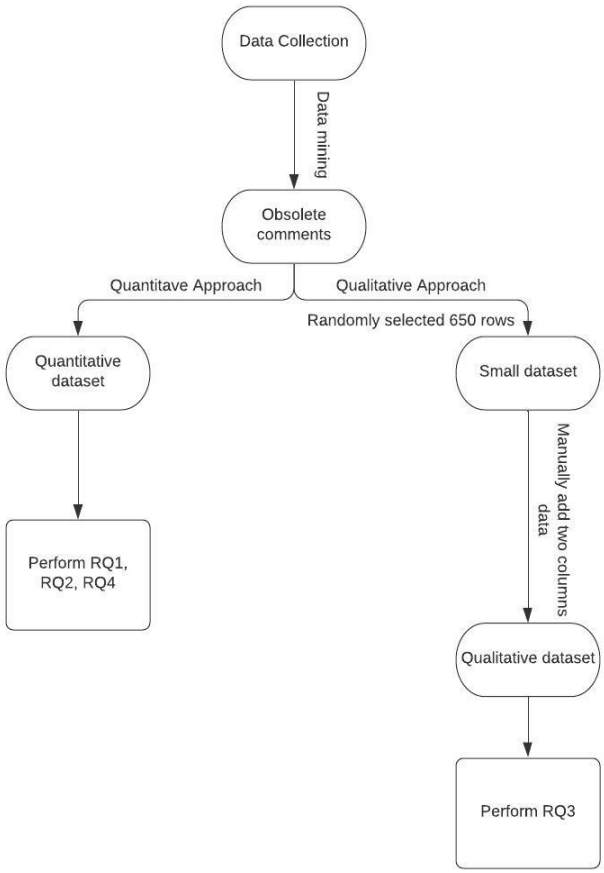

Figure 9. Flowchart of our analysis for every RQ

In this figure, we described the flow of performing different RQs. At the beginning, for performing the different tasks on the data we needed the dataset so we gathered the Stack Overflow data for 2008 to June 2021, and generated the huge dataset of Stack Overflow comments. That dataset is not compatible for performing our different RQs that is why we did a data mining process on that collected dataset and generated the relevant dataset. Now there are two approaches in it to perform the different RQs. For example, for performing RQ1, RQ2 and RQ4 we used qualitative approach and for performing RQ3 we performed qualitative approach. When we talk about qualitative approach, we used the whole dataset and then we achieved the results of RQ1, RQ2, and RQ4. While in RQ3 we used a qualitative approach that means in this approach we selected a few rows from the dataset instead of the whole dataset, and the primary point was that 650 rows selected were random. In this selected 650 rows, we added two more columns and then we added the data manually in these two columns by reading the comments and generating the qualitative dataset. This dataset helps to perform the RQ3.

6 Related work

As reference we reviewed below related work to understand the previous result.

6.1 API deprecation

Using a deprecated API is no longer acceptable due to API updates. Although outdated classes, functions, and variables are still being used, they may be phased out in the future [17]. API deprecation has been the subject of numerous investigations. For example, Jing Zhou and colleagues investigate the use of API deprecation in 26 open source Java frameworks and modules [4]. Suntae Kim et al. the variation of deprecated APIs in nine java-based libraries and find out supporting information for obsolescences [1]. On Stack Overflow, G. Bavota et al. looked into how developers react to the deprecation of Android APIs [5]. API obsolescence is typically caused by the Swift programming language's quick development cycles, according to Pinto et al [6].

We observed all earlier research related API obsolescence, and we observed that all papers focused on API deprecation due to several reasons like version changes, security updates and many more. So, which causes issues in the current system. But we focus on all other reasons which cover obsolescence in the answers on Stack Overflow.

6.2 Obsolete comments from stack overflow

For the analysis and implementation of methods to observe the obsolete answers on stack overflow we go through with Zhang et al.'s research paper. In this paper they observed all RQs with a dataset from 2008 to 2018. But in our analysis we observed obsolete answers in each RQ from 2008 to 2021 dataset. So, our result is quite different from this research paper. For example, the ratio of obsolete answers in third party libraries is 34.5% in our research but in a given research paper it shows 31.7%. That means with evolving time preference of users demand for asking questions and commenting answers will change and this will affect the result.

This research paper performed research on quantitative and qualitative analysis where each analysis performed in every where its demand. So, we also follow this approach in our research. In a qualitative approach, researchers focus on more than 650 obsolete answers manually, which helps them to identify the issue greatly.

6.3 Replicated Questions on stack overflow

Stack Overflow is a popular destination for software developers looking for answers to their code-level difficulties. As a result, they usually include sample code segments and issue descriptions with their questions. Unfortunately, reproducing their stated difficulties from such code parts is not always achievable. Saikant et al. manually observe 400 randomly selected questions from stack overflow and find out the option for reproducibility by using code segments [7]. The researchers next look at the relationship between issue (question) reproducibility and answer meta-data. Their results suggest that question repeatability is likely to generate a more broad range of functions, including acceptable answers. Only 3.89 percent of 914,974 Java code samples on Stack Overflow are parsable, and only 1.00 percent are compilable, according to Yang et al [8]. They parse and compile the code using automated tools such as Eclipse JDT and ASTParser, and they only look at code segments contained in accepted Stack Overflow responses. Rahman and colleagues look into why questions on Stack Overflow remain unanswered [9].

Reason behind choosing this study to observe the pattern in question replication, which helps us to understand what happens when answers are observed as obsolete from the user side. So, after identifying the obsolete content in answer how users replicate their answers by edit or new command. This analysis helps to obtain information about how much time users will take to replicate obsolete answers.

6.4 The effectiveness of QA sites is demonstrating changes in knowledge sharing in Open-Source Software Communities.

B. Vasilescu et al. [10] describe all the information about how the social QA sites did the changes about knowledge sharing in open-source software communities. Here they compared the old community support (mailing lists) and the new community support (StackExchange network) with applying some tasks for different RQs. They describe the drastic change in it and that change happened with the help of gamification like in StackExchange users are trying to collect badges and reputation points for competing others.

For determining the results, they primarily use the case study of R which is widely used for data analytics. The types of talks that occur in mailing lists, the types of questions posed, and the types of responses given were researched qualitatively by Singh et al. [11] and Guzzi et al. [12]. Bettenburg et al. discovered issues that arise when processing mailing list data using off-the-shelf approaches [13]. Through this approach we can easily identify how actively people are involved in this type of website and with the help of this information we can get an idea of how to overcome the obsolete answers problem so at the end we will get a good resultant graph.

7 Conclusion

During the analysis of an empirical study of the obsolete knowledge on Stack Overflow research paper, we observed that there are several approaches applied to identify the obsolete answers from the comments. During our research, we get the result that more than 50% of answers were observed by users on the very first day, which saves other users from obsolete content. To handle obsolete answers issues, Stack Overflow should be a focus on such an innovative system, which helps users to identify the obsolete answers while they are posting. Another thing is answerers are only allowed to answer with a valid version of knowledge in the present time in the comments. In RQ2, tags like node.js, swift, and android are more likely to have obsolete answers to their questions on the post. In the qualitative analysis, we observed that third-party libraries and programming knowledge are containing a high volume of obsolete content. Lastly, all other commenters are most likely to observe obsolete answers in the comments and their ratio is around 51.4%.

8 Future work

Stack Overflow data is vast in size, and it is gradually increased every single year due to users' search on the website. The reason behind this large volume of the dataset is an action performed by users like answering on the post, posting queries as a form of questions on the post, upvotes, downvotes, and many more. So managing this large volume of dataset with an algorithmic tool with a machine learning approach or simply a basic search strategy might be developed to detect existing old answers on Stack Overflow or to assist answerers in recognizing outdated answers in real-time as they respond. Furthermore, in the future, we will be able to thoroughly examine the dataset and find a new approach for analyzing the obsolescence answers from the comments, similar to the above four research topics.

9 Acknowledgments

We express our sincere gratitude to Prof. Dr. Muhammad Asaduzzama of LAKEHEAD UNIVERSITY'S MS in the Computer Science department for his support and direction on this project, as well as the time and effort he put into assisting us in completing the project successfully. This is to express our deepest appreciation and admiration for the lecturer. We would like to express our gratitude to the faculty and department heads at LAKEHEAD UNIVERSITY for their assistance.

Finally, we would like to express our sincere thanks to everyone who has directly or indirectly assisted us in making our initiative a success.

References

[1] Ko D, Ma K, Park S, Kim S, Kim D, Le Traon Y. Api document quality for resolving deprecated apis. In2014 21st Asia-Pacific Software Engineering Conference 2014 Dec 1 (Vol. 2, pp. 27-30). IEEE.

[2] Zhang, H., Wang, S., Chen, T-H., Zou, Y., and Hassan, A.E., "An Empirical Study of Obsolete Answers on Stack Overflow", IEEE Transactions on Software Engineering, Volume: 47, Issue: 4, April 1 2021

[3] Wu Y, Wang S, Bezemer CP, Inoue K. How do developers utilize source code from stack overflow?. Empirical Software Engineering. 2019 Apr;24(2):637-73.

[4] Zhou, J., Walker, R.J., "API deprecation: a retrospective analysis and detection method for code examples on the web", November 2016, the 2016 24th ACM SIGSOFT International Symposium.

[5] Linares-Vasquez, M., Bavota, G., Di Penta, M., Oliveto, R., and Poshyvanyk, D., "How do API changes trigger Stack Overflow discussions? A study on the Android SDK," in Proceedings of the 22nd International Conference on Program Comprehension, ser. ICPC '14, 2014, pp. 83–94.

[6] Rebouas, M., Pinto, G., Ebert, F., Torres, W., Serebrenik, A., and Castor, F., "An empirical study on the usage of the swift programming language," in 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), vol. 1, 2016, pp. 634–638.

[7] Mondal, S., Rahman, M.M., Roy, C., "Can Issues Reported at Stack Overflow Questions be Reproduced? An Exploratory Study".

[8] Yang, D., Hussain, A., and Lopes, C. V. "From query to usable code: an analysis of stack overflow code snippets", In Proc. MSR, pages 391–402, 2016

[9] Rahman, M. M. and Roy, C. K., "An insight into the unresolved questions at stack overflow" In Proc. MSR, pages 426–429, 2015.

[10] Vasilescu B., Serebrenik A., Devanbu P. and Filkov V. (2014). How social QA sites are changing knowledge sharing in open source software communities. Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work Social Computing - CSCW '14. https://doi.org/10.1145/2531602.2531659

[11] Singh, V., Twidale, M. B., and Nichols, D. M. Users of open source software - How do they get help? In Proc. HICSS, IEEE (2009), 1–10.

[12] Guzzi, A., Bacchelli, A., Lanza, M., Pinzger, M., and van Deursen, A. Communication in open source software development mailing lists. In Proc. MSR, IEEE (2013), 277–286.

[13] Bettenburg, N., Shihab, E., and Hassan, A. E. An empirical study on the risks of using off-the-shelf techniques for processing mailing list data. In Proc. ICSM, IEEE (2009), 539–542.

[14] "Stackexchange data explorer", 2021, Available at: https://data.stackexchange.com/

[15] Available at: http://socialcompare.com/en/comparison/ android-versions-comparison

[16] "Stack Exchange Data Dump", by Stack Exchange, Inc., Available at: https://archive.org/details/stackexchange

[17] "Deprecation of APIs", 2020, Available at: https://docs.oracle.com/javase/7/docs/technotes/guides/javadoc/deprecation/index.html

[18] "Stack overflow dataset description", Available at: https://meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede