Stack Overflow: A Code Laundering Platform?

Le An, Ons Mlouki, Foutse Khomh, and Giuliano Antoniol
SWAT–SOCCER Labs, Polytechnique Montréal, Québec, Canada
{le.an, ons.mlouki, foutse.khomh, giuliano.antoniol}@polymtl.ca

***Abstract*—Developers use Question and Answer (Q&A) websites to exchange knowledge and expertise. Stack Overflow is a popular Q&A website where developers discuss coding problems and share code examples. Although all Stack Overflow posts are free to access, code examples on Stack Overflow are governed by the *Creative Commons Attribute-ShareAlike 3.0 Unported* license that developers should obey when reusing code from Stack Overflow or posting code to Stack Overflow. In this paper, we conduct a case study with 399 Android apps, to investigate whether developers respect license terms when reusing code from Stack Overflow posts (and the other way around). We found 232 code snippets in 62 Android apps from our dataset that were potentially reused from Stack Overflow, and 1,226 Stack Overflow posts containing code examples that are clones of code released in 68 Android apps, suggesting that developers may have copied the code of these apps to answer Stack Overflow questions. We investigated the licenses of these pieces of code and observed 1,279 cases of potential license violations (related to code posting to Stack overflow or code reuse from Stack overflow). This paper aims to raise the awareness of the software engineering community about potential unethical code reuse activities taking place on Q&A websites like Stack Overflow.**

***Index Terms*—Software licenses, Stack Overflow, Q&A website, Knowledge repository, Mining software repositories**

I. Introduction

Question and Answer (Q&A) websites, such as *Stack Overflow*[1], allow users to share knowledge and expertise. These websites have become large knowledge repositories for developers to communicate on technical problems and resolve programming issues. However when reusing code from Stack Overflow, developers should comply with the license of the code. Software licenses govern the use or redistribution of software. A failure to comply with a license term can result in bitter legal battles and large fines, as evidenced by the legal battle between Google and Oracle over nine lines of code [1]. Stack Overflow applies the *Creative Commons Attribute-ShareAlike* (CC BY-SA 3.0) license [2] to restrict the usage of its content. Therefore, developers who violate the terms of this license when reusing code from Stack Overflow are exposed to penalties. Also, if a developer copies code from an existing software system and shares it on a Q&A website (like Stack Overflow) without citing the reference, she would also be violating the software's copyright terms. Copying code from Stack Overflow to a software system or the other way around can lead to license violations and developers could be sued by the code owners.

In a previous work, Sojer et al. [3] observed that developers do not always check copyright terms thoroughly when reusing code from Internet accessible open-source software. They also observed that some developers intentionally ignore the obligations imposed by licenses when reusing code from open-source software [3]. In a recent study [4], we found 17 Android apps with license violations; suggesting that the developers of these apps disregarded the legal constraints of licenses' terms when reusing code from third-party sources in their software. However, both of these studies did not investigate the role that Q&A websites could have played in these license violations. Yet, copy-paste operations from (and to) Stack Overflow can also lead to license violations. In particular, although Stack Overflow is free to access and its content can be easily searched by Google, developers seem to have less knowledge about the restrictions of Stack Overflow, in comparison to other software systems; as illustrated by this discussion on Stack Exchange [5].

In this paper, we conduct a quantitative study to investigate whether developers respect license restrictions when reusing code from Stack Overflow to Android apps, or posting the code of an Android app in a Stack Overflow question. We analyze 79.2k files extracted from 399 apps and 2.1M Stack Overflow posts that are related to Java and Android questions. We use a state-of-the-art clone detection tool [6] NiCad [7], to identify duplicate code between the two studied datasets (*i.e.*, Apps' code and Stack Overflow posts). To ensure that code clones reported by NiCad are real code clones, we manually validate all occurrences of code clones found between the two datasets. We answer the following four research questions:

RQ1: Do developers release apps with code copied from Stack Overflow?

In the 399 subject apps, we found 232 Android code snippets that are exact clones of code snippets posted on Stack Overflow. These code snippets are distributed in 135 files from 62 different apps. This result provides a quantitative evidence of potential code copying from Stack Overflow to Android apps.

RQ2: Do developers respect the copyright terms of code reused from Stack Overflow?

We investigated the licenses of the 232 code snippets that were potentially reused from Stack Overflow, and observed potential cases of license violations in 60 apps. We contacted the developers of the apps in which the violations were found and received some confirmation of code reuse from Stack Overflow, with one developer saying: *"there is definitely code in our project that is copy-pasted from Stack Overflow, as I have done this*

[1] http://stackoverflow.com

several times. I assumed (falsely it seems) that everything there is public domain". These results are an indication of potential unethical code reuse from Stack Overflow. Software organizations should consider putting in place license control and management mechanisms to avoid exposing themselves to license violation issues.

RQ3: Do Stack Overflow users respect copyright terms when publishing code snippets on Stack Overflow?

We observe 1,226 Stack Overflow posts potentially reusing code from respectively 68 Android apps. 1,219 of these posts have a potential risk of license violations. A majority (83.9%) of the large code snippets (with more than 50 lines) contained in these Stack Overflow posts are related to the *Android Navigation Drawer* component. We also found 126 code snippets that seem to have migrated from one app to Stack Overflow and then from Stack Overflow to another app. In 12 of the migrated code snippets, the file containing the code snippet in the first app and the file containing the code snippet in the second app use different software licenses.

RQ4: How long does a Stack Overflow code snippet remain in released versions of an app?

Most of the code reused from Stack Overflow remained in the apps for up to 20 releases. In some cases, the code remained in the app for more than 300 releases and–or during a period of more than four years. The fact that these code snippets with potential license violations remained in the apps for such a long time suggests that some developers do not pay enough attention to copyright terms.

Overall, this paper makes the following contributions:

- To the best of our knowledge, this is the first quantitative study about the misuse of software licenses on a large Q&A website.
- We provide quantitative and qualitative evidences of potential unethical code reuse on Stack Overflow. We hope that the results of this study will raise the awareness of the software community about license issues in Q&A websites, which are now very popular in developers' communities.

The remainder of this paper is organized as follows. Section II discusses license restrictions in open-source software and Stack Overflow. Section III describes the design of our case study. Section IV presents the results of the case study. Section V discusses threats to the validity and the contributions of this study. Section VI summarizes related works and Section VII concludes the paper.

II. License Restrictions in Open-Source Software and Stack Overflow

In this section, we discuss general license restrictions in open-source software and Stack Overflow. Open-source software licenses allow free access to the source code of a software system. However, reuse and–or distribution are often limited by certain restrictions [8]. Most open-source software licenses possess different versions, each of them with their own restrictions. There exist two kinds of open-source licenses: *restrictive* licenses (also known as "copyleft" or "reciprocal" licenses) and *permissive* licenses [9]. Restrictive licenses enforce restrictions on the license of derivative works. For example, the GPLv3.0 license says this about derivative works: *"You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy"*. However, permissive licenses allow software distribution under a different license (*e.g.*, BSD and MIT licenses).

In Stack Overflow, all user-generated content is licensed under the *Creative Commons Attribute-ShareAlike 3.0 Unported* license (CC BY-SA 3.0) [2]. Under this license, users can share and adapt the content in the website, but they must respect the following restrictions [10]:

- Attribution: *"You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests that the licensor endorses you or your use."*
- ShareAlike: *"If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original."*

In general, if a user copies a code snippet from Stack Overflow and reuses it into her own project, she must provide a reference to the original material and a link to CC BY-SA 3.0. She should also indicate any changes in case of a derivative. In addition, she can only share or release these projects under the CC BY-SA 3.0 or its later versions.

III. Case Study Design

In this section, we describe the data collection and analysis approaches that we use to answer our four research questions.

Figure 1 shows a general overview of our data processing approach. We describe each step in our data processing approach below. The corresponding data and scripts are available online at: https://github.com/swatlab/stack_overflow.

A. Data Collection

In our previous work [4], we found 399 apps with license inconsistences (*i.e.*, files that share similar code but having different licenses). In this paper, we leverage this dataset of 399 apps to investigate the role that the Q&A website, Stack Overflow, could have played in the occurrence of these license inconsistencies. We focus on files with license inconsistencies because they are likely to cause license violations. In total, the 399 apps contain 79,222 files with inconsistent licenses, which account for 1.4GB. We intend to investigate whether these files contain any code snippet reused from–or–to Stack Overflow. Stack Overflow shares its data in XML format as part of the Stack Exchange data dump[2], which is updated every three months under the CC BY-SA 3.0 license [2]. In this paper, we study Stack Overflow's data dump from July 2008 until March 2016.

[2] https://archive.org/details/stackexchange

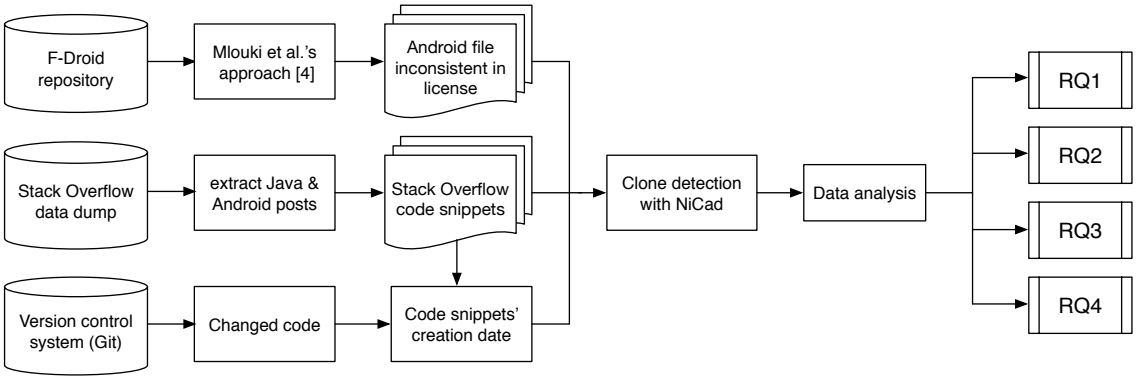

Figure 1: Overview of our data processing approach.

B. Preprocessing of Stack Overflow Data

Stack Overflow's posts are stored in the `Posts.xml` file of the Stack Exchange data dump. `Posts.xml` accounts for 42GB. We write a Python script to identify posts from this file. From each post, we use regular expressions to extract the following information:

- *Post ID*: the identifier of a post. Different posts in the same discussion thread possess different IDs.
- *Post creation date*: the submission date of a post.
- *Post tags*: the topics of a discussion thread, such as programming languages (*e.g.*, Java, C++), development environments (*e.g.*, Eclipse), and deployment platforms (*e.g.*, Android, iOS).
- *Post body*: the content of a post, which keeps the HTML format as in Stack Overflow's website.

In this paper, we only study posts with `Java` and `Android` tags, because we will compare the similarity of code snippets extracted from Stack Overflow with the source code of Android apps, and the majority of code in Android apps is written in Java. We use the creation date of Stack Overflow posts to decide whether a code snippet appeared on Stack Overflow before or after the creation of its corresponding clone in an Android app. If a code snippet was posted on Stack Overflow before the apparition of its clone in an Android app, we consider it as a reuse candidate from Stack Overflow to the Android app, meaning that the developers of the Android app probably reused the code from a Stack Overflow post. If the code snippet appeared in the App first before it was posted on Stack Overflow, we consider it to be a code reuse candidate from the Android app to Stack Overflow. Since the body of a post is kept in HTML format, we use the following regular expressions to extract source code snippets from the post:

```
<pre><code>(.+?)</code></pre>
```

We save each extracted code snippet into a separate file. We eliminate the snippets with less than 10 lines of code, because too few lines of code can lead to noises in clone detections. In total, 2,106,303 code snippets are considered in our study, which account for 8.6GB.

C. Clone Detection

We use the clone detection tool, NiCad [7], to identify duplicate code between the studied source code datasets, *i.e.*, Android app dataset and Stack Overflow dataset. NiCad can detect *Type 1* (exactly similar code snippets), *Type 2* (syntactically similar code snippets), and *Type 3* (copied code with further modifications) clones [11]. It can handle source code written in multiple languages, such as Java, C++, and Python. Svajlenko et al. [6] compared the performance of 11 clone detection tools from the literature and reported that NiCad achieved higher precision and recall, in comparison to the other 10 clone detection tools. In addition, NiCad's cross-project clone detection feature allows us to only detect code clones between the two datasets instead of within each dataset. Since both studied datasets are very large in size and clone detection is a very resource consuming process, the cross-project clone detection feature of NiCad is useful to reduce the cost of the clone detection process, allowing us to analyze large code repositories. We use the default settings of NiCad, *i.e.*, each clone pair has more than 70% of similarity and the clones contain at least 10 lines of code.

During the clone detection process, NiCad requires an analytic memory space whose size is often 50 times larger than the analyzed dataset. Considering the size of our two datasets (*i.e.*, Android apps (79.2k files, 1.4GB) and Stack Overflow code snippets (2.1M files, 8.6GB)), we cannot feed the whole dataset into NiCad. Consequently, we split both datasets into slices. We limit the size of each Stack Overflow slice to 2,000 code snippets. Thus, the Stack Overflow data are split into 55 subsets. Each subset accounting for 160MB in average. Similarly, we split the Android dataset into 100 subsets, where each subset accounts for 14MB in average. We deliberately set each Stack Overflow slice larger than each Android slice, because firstly, NiCad will automatically filter out some irrelevant code such as code not corresponding to the syntax of Java. In addition, we tuned the split number for both studied datasets, the current splitting strategy allows NiCad to provide results faster.

Next, we perform clone detection with NiCad for $55 \times 100 = 5,500$ rounds. We write a Python script to automate these clone detection rounds, *i.e.*, when one round is finished, the

next round will be automatically started. Finally, we combine the results of each of the subsets as the total results of the clone detection between the two studied datasets. In this paper, we leveraged multiple computers with 32GB or 64GB memory, and finished the whole clone detection process in more than one month (including reprocessing for some failed clone detection rounds).

IV. Case Study Results

This section presents and discusses the results of our four research questions. For each question, we describe the motivation, the approach followed to address the question, and the findings. To simplify the text, we define the following terms:

- App: one of our studied Android applications.
- Post: One of our studied Stack Overflow post. We refer to the author of a post as a poster.
- Similar: Two pieces of code are *similar* if they are identified as being clones by NiCad (with its default parameters).

RQ1: Do developers release apps with code copied from Stack Overflow?

Motivation. Stack Overflow allows developers to ask and answer questions about programming problems [12]. If a piece of code from Stack Overflow addresses a developer's issue, she may reuse the code in her project (sometimes with little modification). In this preliminary question, we look for evidences of code reuse from Stack Overflow in Android apps. We are interested in understanding the role that Stack Overflow could have played in the occurrence of the license inconsistencies observed in our previous study [4].

Approach. As described in Section III-C, we use NiCad to identify code clones between Stack Overflow posts and Android app files. For a given clone pair, if the Android code was created later than the Stack Overflow code, we consider that the cloned code was reused from Stack Overflow to the Android app and flag the Android code as a "code reuse candidate". To identify the creation date of a clone snippet from an app's file, we compare the clone snippet against the whole revision history of the file. We write a Python script to automatically match a cloned snippet to each *added* line in the corresponding file's changing commits in Git. We note the date of the earliest matched commit as the creation date of the code snippet from the Android app. In total, we found 434 Android code snippets that are *similar* to a Stack Overflow code snippet. 346 of these Android code snippets' creation date can be automatically identified. For the remaining 88 Android code snippets, we manually reviewed their files' revision history to identify the creation dates of the cloned code snippets.

To determine whether a code snippet reused from Stack Overflow can lead to a license inconsistency, we proceed as follows. First, we find the snippet's corresponding file and locate the snippet's line numbers ($Line_{clone}$) in the file. Next we identify the line numbers ($Line_{inconsist}$) of the portion of code in this file, which is concerned by the license inconsistency (following the same approach as in our previous work [4]). Then, we calculate the common line numbers ($Line_{common}$) in $Line_{clone}$ and $Line_{inconsist}$. Finally, we compute the rate of $Line_{common}$ over $Line_{inconsist}$. We refer to this rate as *overlapped rate* in the rest of the paper. If the *overlapped rate* is greater than 0, it means that the reused candidate (or a part of the reuse candidate) is concerned by the file's license inconsistency issue; implying that this reused code can lead to a software license violation issue.

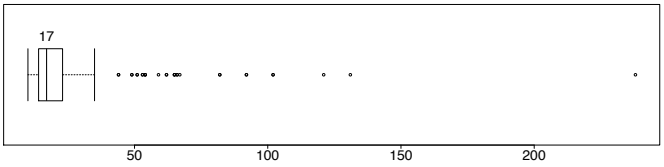

Figure 2: Number of lines of an Android code snippet similar to a Stack Overflow post.

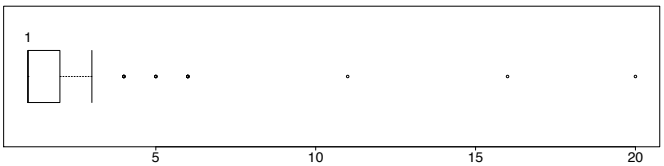

Figure 3: Number of Android code snippets similar to the same Stack Overflow post.

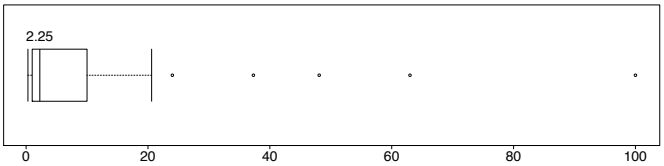

Figure 4: *Overlapped rate* (in %) for each code snippet that is similar to a Stack Overflow post.

Findings. We found that 232 Android code snippets are *similar* to the code in Stack Overflow posts, and the Android snippets were created later than the corresponding posts. These code snippets are distributed in 135 files from 62 different apps. In other words, it is very likely that 15.5% of the studied apps have reused code from Stack Overflow. Figure 2 shows the distribution of the sizes (in terms of numbers of lines of code) of the Android code snippets that were potentially reused from Stack Overflow. The median number of lines of code in these reused code snippets is 17, which is slightly higher than NiCad's minimum line number for clones (*i.e.*, 10). This result suggests potential code copying from Stack Overflow to Android apps. It also shows that when developers reuse code from Stack Overflow, they take only few lines. Nevertheless, we found 25 reused code snippets with a size of more than 50 lines of code. We manually checked these large code snippets and found that:

- The author of the post (#23349354) shared multiple classes on about the refreshment of the Android ListView. The CLOVER app has several methods highly *similar* to the code contained in that post. We searched for the name of the author of this post in the app's contributor list on

Github, but did not find any match. The Stack Overflow post was created in April 2014, while the corresponding Android code snippet was introduced in the app more than 10 months later (January 2015). Hence, it is very likely that the code snippet was reused from the Stack Overflow post by a developer of the app.

- A developer posted a blur animation function on Android ListView (#23844259). A *similar* code snippet with more than 100 lines appeared later in the app ACDISPLAY. As in the previous case, we could not find the name of the author of the post among the contributors of the app. Also, the code was introduced in the app more than four months after it was posted on Stack Overflow. It is very likely that the developers of the app reused code from the post.
- The OPACCLIENT app has a whole class `FlowLayout` highly *similar* to the code contained in post #16761418. We could not find the poster name in the list of contributors of the app. The post was created in May 2013, more than nine months before the corresponding code snippet was introduced in the app (in March 2014). Again, it is very likely that the code was reused from the post to the app.
- Two releases of the CLOVER app contain several methods that are *similar* to the code contained in post #21857260. The post is a question about the InflateException of a custom view. It was submitted only one day before the corresponding code was introduced into the app, then the post author explained that he resolved the problem himself in the same day of the question. Therefore, we cannot confirm this case as a copy-paste operation from Stack Overflow to an app. A possible scenario could be that a developer posted some uncommitted code on Stack Overflow as a question. When the problem was resolved (by himself in this case), he applied the code in the app and committed it.

The topic of 19 out of the 25 code reuse candidates with more than 50 lines is about Android user interface. We observe that developers post problematic code on Stack Overflow when they encounter issues (such as crashes). To help other programmers understand and debug the code, the posters tend to share full classes, which also allows other developers to reuse the code to their own projects.

> *Android user interface (UI) is a hot topic on Stack Overflow. Developers reuse large UI related code snippets from Stack Overflow posts to their apps. These code snippets often contain the whole functionality of some classes.*

If we group the 232 code reuse candidates by their origins (*i.e.*, the original Stack Overflow posts), there are 45 groups where more than one candidates are related to the same Stack Overflow post. Figure 3 shows the distributions of the number of Android snippets (noted as $N_{Android}$) that are *similar* to the same Stack Overflow post. Overall, the median number of snippets *similar* to a Stack Overflow post is 1; implying that most of the Android code snippets were reused from different posts. We manually analyzed the outliers (*i.e.*, $N_{Android} > 3$) in Figure 3, and summarized the typical findings as follows:

- Several methods in the app, WALLETCORDOVA, were *similar* to the code from post #21907131. The *similar* code is used for handling the File-transfer plugin in PHONEGAP 3.3. The code in the Stack Overflow post and in the app is almost identical with only few modifications on variable names. It is very likely that the app's developers copied code from the post.
- The method `copyFile` contained in ANKI-ANDROID files is *similar* to the code contained in post #7269278. However, we did not find the author name of the post in the list of developers of the app file, or in the list of contributors to the app. The "file copying" method is not originally provided by Java. Thus, it is very possible that the developers of the app reused code from the post.
- The app, FROSTWIRE-ANDROID, possesses several methods *similar* to code in post #20027718. The poster of #20027718 shared a whole class that can be used to customize the class `IconPageIndicator`. The app's class has multiple methods in common with the post's class, but also has some new methods that are not provided in the post. The common methods are perfectly identical. Some of the *similar* methods have less than 10 lines of code, and hence were not identified by NiCad as clones, but we could identify them during our manual analysis. These similarities are a strong indication that developers may have reused code from that post to their app.
- FROSTWIRE-ANDROID, OPENLAW, and READER have a common method `setCurrentItem` which is *similar* to a code snippet from the post #14433281; implying that this method was modified and reused into different apps.
- Several methods in the post #8327136 have their *similar* counterparts in FROSTWIRE, OPENLAW, READER, K-9, TASKS, TRANSDROID, QUASSELDROID, AD-AWAY, and ATOMIC. The methods were introduced in the apps later after the creation of post #8327136. This "popular" post shows an example of creating a horizontal ScrollView in the Android Fragment. It is very likely that the developers of these apps borrowed code from the post.

Overall, we observe that most of the code reused candidates were reused in a single app. Most these reused code candidates concerned general purpose issues, *e.g.*, how to set visual components for an Android app. Although our clone detection tool, NiCad, is set to identify only code clones that are equal or larger than 10 lines of code, we manually found some small Android code snippets (less than 10 lines of code) that are identical to the code in a post. Some long posts were reused by multiple apps or by multiple files in the same app.

Figure 4 shows the *overlapped rate* (in %) for each of the 232 code snippets that are similar to a code snippet in a Stack Overflow post. In this figure, we only depict rates

that are greater than 0, which is found in 88 code snippets. The median *overlapped rate* is 2.25%. In other words, only few lines of code are both *similar* to a Stack Overflow post and are contained in the license inconsistent range. Only for two code snippets, the *overlapped rate* is greater than 50%. However, few lines with copyright violations are enough to expose an organization to penalties.

RQ2: Do developers respect the copyright terms of code reused from Stack Overflow?

Motivation. Stack Overflow allows developers to reuse its content. But developers must respect the restrictions of the *Creative Commons Attribute-ShareAlike 3.0 Unported* (CC BY-SA 3.0) license. Briefly, when reusing code from a Stack Overflow post, developers must cite the reference of the original post and license the resulting derivative work under CC BY-SA 3.0 or its later versions. They also need to indicate the changes if they modified the original code. In **RQ1**, we have found 135 Android files (from 62 apps) that potentially reused code from Stack Overflow. In this research question, we intend to investigate whether the developers of these apps respected license restrictions when "reusing" code from Stack Overflow.

Approach. We manually analyze the license declaration of the Android files containing code that we believe were cloned from Stack Overflow. We examine (1) whether these files use the CC BY-SA 3.0 or its later versions; (2) whether developers use CC BY-SA 3.0 or its later versions in the apps' main licenses; and (3) whether they cite the reference of the original Stack Overflow posts. If any of these conditions is not satisfied, we consider that the corresponding developers did not respect copyright terms. To validate the results of our quantitative analysis, we send an anonymous survey to the developers of the apps that we consider as violating Stack Overflow's license. We limit the survey to apps that contain code snippets with a more than 90% similarity (with a code snippet from Stack Overflow). We ask developers (1) whether the code snippets that we found were effectively reused from Stack Overflow, (2) whether they often reuse code from Stack Overflow, and whether they (3) consider Stack Overflow to be a reliable source of information.

Findings. None of the 135 files that contain code potentially reused from Stack Overflow were released under CC BY-SA 3.0 or its later versions. And none of these files contains a reference to the corresponding Stack Overflow post (that contains the *similar* code snippet). We found two posters' names in the list of contributors of the corresponding apps; indicating that the developers may have copied code from their own posts to the apps. The remaining 60 apps are therefore at risk of license violations. We contacted 23 developers working on these apps and received six answers. All the six developers who replied confirmed that they copied code from Stack Overflow to their projects, with one of them saying: *"there is definitely code in our project that is copy-pasted from Stack Overflow, as I have done this several times. I assumed (falsely it seems) that everything there is public domain ... If I were to never look at code examples, and only write code from reading the APIs, I would probably miss elegant solutions and overlook important pitfalls."*. Another developer said: *"I often turn to StackOverflow for coding solutions ... I publish my code snippets there also ... (regarding code reuse from Stack Overflow) I would say that using code snippets 'as is' usually is impossible/impractical."* Regarding the specific code snippet that was asked, one developer replied that : *"I don't remember copy-pasting code from other sites ... I actually inherited it (the cloned code) from (another project) ... I don't oppose copy-pasting to be very honest. If it was just a code snippet and I understand it, and it does what I want, I would copy-paste it to my code"*.

These results show that developers often turn to Stack Overflow for solutions. Some of them believe that reusing code from a Q&A website like Stack Overflow can improve the quality of their software. One developer even suggested that Stack Overflow updates its license to CC BY-SA 4.0, in order to be compatible with the GPL license: *"(Regarding Stack Overflow's license) it appears to not be compatible with the GNU-GPL ... I hope the staff at Stack Overflow will address the problem"*. One of the developers lamented the lack of policy about licenses in his organization: *"we don't have any policy about that. Now might be a good time to have that discussion ... I have copy-pasted from Stack Overflow in the past, and still do it on projects I work on, usually with a comment citing the Stack Overflow URL"*.

> *We recommend that software organizations put in place license control and management mechanisms, to avoid exposing themselves to license violation issues.*

RQ3: Do Stack Overflow users respect copyright terms when publishing code snippets on Stack Overflow?

Motivation. In **RQ1** and **RQ2**, we have found evidences of code reuse from Stack Overflow to Android apps, with potential license violations. In this question, we investigate whether developers use code from Android apps to ask or answer questions on Stack Overflow and whether they respect copyright terms when doing so. We also want to know whether there are cases of code migration to and–then from Stack Overflow, a phenomenon that we refer to as "code laundering" because the original license of the code would be altered by a transit on Stack Overflow.

Approach. In the cloned pairs found between Stack Overflow code snippets and Android code snippets, we identify the Stack Overflow snippets that were posted later than their corresponding Android snippets' creation date. We consider that these Stack Overflow snippets reused code from an Android app. If a Stack Overflow post reused code from an app without providing the app's license, we consider it to be a license violation. We use a Python script to automatically detect license information in the corresponding posts and manually validate our results. In addition, for each code snippet reused from an app to Stack Overflow, we examine whether the code

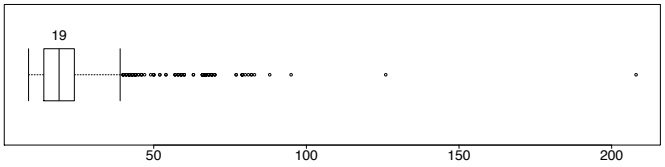

Figure 5: Number of lines of code cloned from Android apps to Stack Overflow.

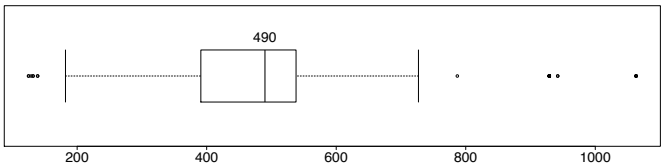

Figure 6: Duration (in days) of code migration from one app to another app.

snippet was reused later into another Android app. We also compare the software licenses of the two apps (*i.e.*, the source and the destination after the transit on Stack Overflow) and check their consistency.

Findings. We found 1,226 Stack Overflow posts containing code snippets that were reused from 68 Android apps. However, only five of the posts provide the license of the original code. Although some posters claim that the code is from their own projects, we can only match two poster names in the apps' contributor lists. There is therefore a risk of license violation in the remaining 1,219 posts. Figure 5 shows the distribution of the sizes (in terms of numbers of lines of code) of the Stack Overflow code snippets that were potentially reused from Android apps. The median size of these code snippets is 19 lines of code. However, we found 112 code snippets with a size of more than 50 lines of code. To understand why some developers reused such large code snippets from Android apps to Stack Overflow, we manually examined the content of the 112 posts that contain the code snippets, and made the following observations:

- 107 out of the 112 code posts are related to Android UI design, with 94 posts ($94 \div 112 = 83.9\%$) focusing on *Android Navigation Drawer* [13], which is a panel that displays the app's main navigation options on the left edge of the screen [13]. This result is surprising, because previous work (such as [12]) did not report the Navigation Drawer as a hot topic of Android. Also, in **RQ1**, we did not observe a large number of code reuses related to the Navigation Drawer from Stack Overflow to the Android apps. One explanation could be the fact that code snippets about Navigation Drawer were found only in question posts, in which it is likely that they are used as illustrations for a problem and not solutions.
- Only 3 out of the 112 posts are related to general Java problems. Posts #29242197 and #29154598 discuss the implementation of a `java.util.Comparator`. And posts #29242197 and #28177863 discuss the issue of NullPointerException.
- Two posts are related to other Android problems. Post #21299496 discusses a NullPointerException problem, while post #34858945 discusses a file picker's problem.
- All of the 112 posts are question posts in a discussion thread. In these questions, developers tend to share entire classes to allow other developers to test and debug their problems. However, developers did not provide any license information, in all these posts that potentially reused large code snippets from apps. These posts, which are very likely to violate the copyright terms of the apps, also impede future developers who would like to reuse the code contained in the posts, with the correct license.

> *We found 1,219 Stack Overflow posts with potential license violations. We observe that Android Navigation Drawer is a hot topic in posts that contain large code chunks reused from Android apps. We also observed that developers tend to share entire classes in the question post of a discussion thread.*

In our investigation of potential cases of "code laundering", we found 126 code snippets that first appeared in an Android app, before a code snippet exactly similar to them was posted on Stack Overflow. Later on, an exactly similar code snippet (re-)appeared in another app. We call these code snippets "migrated code snippets". In 12 of the migrated code snippets, the file containing the code snippet in the first app and the file containing the code snippet in the second app use different software licenses. This result shows the risk of migrating code through Stack Overflow. The license of the original code could be altered (during the migration), leading to license violations.

Figure 6 illustrates the duration (in days) of the code migrations (from one app to another app) that were found in our dataset. The median value is 490 days, *i.e.*, these code snippets took in average 16 months to migrate from an app to another app via Stack Overflow. The shortest migration duration is 125 days (4 months), and the longest duration is 1,063 days (35 months).

> *We found 126 code snippets that seem to have migrated from one app to Stack Overflow and then from Stack Overflow to another app. In 12 of the code snippets, the file containing the code snippet in the first app and the file containing the code snippet in the second app use different software licenses. These code snippets spent between 125 to 1,063 days to complete their migration through Stack Overflow.*

RQ4: How long does a Stack Overflow code snippet remain in released versions of an app?

Motivation. In **RQ2** we found code snippets in Android apps that were potentially reused from Stack Overflow. The apps containing these code snippets were not released either under CC BY-SA 3.0 or its later versions. Moreover, we found no

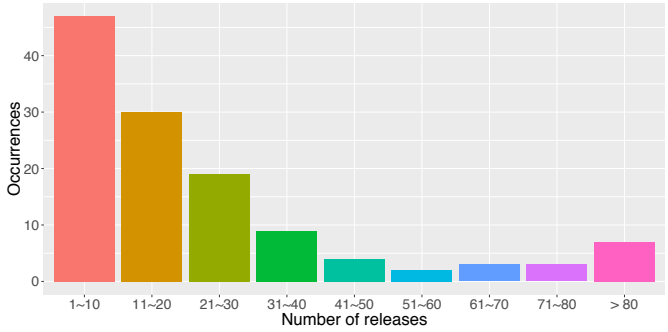

Figure 7: Distribution of the numbers of released versions in which a code reuse candidate remains.

reference to the corresponding Stack Overflow posts (that contain the similar code snippets) in the apps, suggesting potential license violations. In this research question, we examine the lifespan of these code snippets in the apps to understand whether and–how developers address these potential license violation issues.

Approach. For each app containing a code snippet that was potentially reused from Stack Overflow, we track the evolution of the code snippet across the different releases of the app, to identify the first release where the code snippet was introduced and the last release that contained the code snippet. If the code snippet was not removed from the app, the last release containing the code snippet is the latest release considered in our study. We analyze the evolution history of the app from the beginning of the project until February 2015. Instead of considering only the code snippet's corresponding file, we take all files in the app into account. We proceed this way because a code snippet may be removed from one file and reused into another file. We use NiCad to detect duplicate code between the code snippet and each of its app's releases. In **RQ1**, we observed that multiple code snippets in the same app can be *similar* to one Stack Overflow post. These code snippets are also *similar* to each other. In fact, NiCad detects them as a clone class [14]. The 232 code snippets reused from Stack Overflow can be grouped into 124 clone classes. All the code snippets in a clone class belong to the same app. In each clone class, we identify the code snippet with the earliest creation date. We then run Nicad to detect clones between this code snippet and the files of its corresponding app's releases. Since different apps follow different release strategies (*i.e.*, some apps are released more frequently than other apps), we will not only report the number of releases that contain the reused code snippet, but also the duration in days from the introduction of the reused code snippet in the app until its removal or the last day of our study period.

Findings. Figure 7 shows the lifespan (in terms of number of releases) of code snippets reused from Stack Overflow to the apps. Figure 8 shows the number of days during which the code snippets reused from Stack Overflow remained in the apps. Among the 124 code snippets that were tracked through the different releases of the apps, 77 (*i.e.*, 62%) remained in the app for up to 20 releases. 15 code snippets (*i.e.*, 12%)

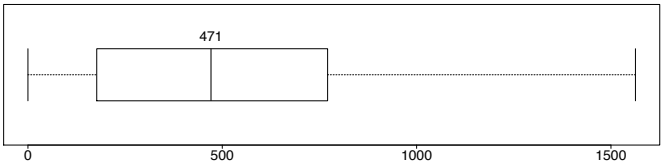

Figure 8: Number of days during which a code reuse candidate remains in an app.

remained in the app for more than 50 releases. We found five code snippets that remained in the apps for only a single release; the developers of these apps may have realized the threat posed by the copied code snippets. However, seven code snippets remained in the apps for more than 80 releases. In `Anki-Android`, we found a code snippet similar to a code snippet from Stack Overflow, that remained for 346 releases.

The median value of the lifespan (in days) of the reused code snippets is 471 days, *i.e.*, around 15 months. The most ephemeral code snippet stayed in its app for only 17 days, and was present in only one release (of `SafeSlinger-Android`). However, we found one code snippet similar to a code snippet from Stack Overflow in `PinDroid`. That code snippet stayed in `PinDroid` for 1,563 days (more than four years) and was present in 30 releases.

Overall, the reused code snippets tend to stay in the apps for a long time, and across multiple releases. This result suggests a lack of awareness from developers, toward the risk of license violations, when reusing code from Stack Overflow.

> *Code snippets reused from Stack Overflow tend to stay in the apps for a long time, and across multiple releases, suggesting that developers do not pay enough attention to copyright terms on Stack Overflow.*

V. Discussion

A. General Discussion

To the best of our knowledge, this is the first quantitative study on license incompatibility between open-source software and a Q&A website. Based on our experience, developers often reuse code from Stack Overflow in their own projects or share their projects' code to Stack Overflow. One motivation for conducting this study was the discussion that we found on Stack Exchange [5] showing developers struggling to interpret the restrictions of the CC BY-SA 3.0 license and exchanging about how to avoid license violations when reusing code from Stack Overflow. No conclusion was drawn from that discussion. A developer suggested to "consult an attorney" on Stack Overflow's "specific legal issues". In this paper, we cannot and do not intend to judge license violations from the findings of our case study. Instead, we aim to raise developers' awareness about the copyright terms on Q&A websites.

Based on the results of our study, when reusing code from a Q&A website, we recommend that developers provide a reference to the original code. Also, whenever it is possible, we suggest that they use a dual license (*i.e.*, both the license

of their project and the website's license) in order to prevent license violations. When sharing code to a website, we also recommend that developers mention the license of the original project from which the code was borrowed and provide a reference to this original project. The reference can also help future developers (who reuse the code) to choose the right software license.

Although this study contains some threats to validity that we will discuss in Section V-B, this paper sheds light on potential unethical code reuse activities taking place on Q&A websites like Stack Overflow. In the future, we plan to study code reuse issues in other Q&A websites and in large-scale open-source systems.

B. Threats to validity

We now discuss the threats to validity of our study following the guidelines for case study research [15].

Construct validity threats concern the relation between theory and observation. In our study, threats to the construct validity are mainly due to measurement errors. We use the state-of-the-art clone detection tool [6], NiCad, to identify similar code between the subject Android files and Stack Overflow code snippets. We use the default setting of NiCad (*i.e.*, minimum clone lines equal or greater than 10) to perform the clone detection, because considering code snippets with too few lines would lead to many false positives. Nevertheless, during our manual analysis, we found some Android code snippets with less than 10 lines of code, that were highly similar or identical to a code snippet from Stack Overflow. However, the goal of this paper is not to report all similar code between the subject Stack Overflow posts and Android files. Instead, we aim to gather some evidences of code reuse activities from Stack Overflow to Android apps (and the other way around), and investigate whether developers respect copyright terms during these code reuse activities. To mitigate noises due to the precision of NiCad, for each research question, we performed manual validations.

Internal validity threats concern factors that may affect a dependent variable and were not considered in a study. Tracking and confirming code reuse from the Internet is a very difficult task. Though we observed some Android code snippets similar or even identical to code snippets on Stack Overflow (including large code chunks), we cannot prove that the code snippets in question were "copied" from Stack Overflow to the apps (or the other way around). Because developers can also reuse code from other websites or open-source systems. However, the developers that we surveyed confirmed the existence of code copying from Stack Overflow to apps, *e.g.*, one developer stated that: *"there is definitely code in our project that is copy-pasted from Stack Overflow, as I have done this several times."* In addition, developers often use pseudo names in their Stack Overflow accounts, which increases the difficulty of deciding whether a developer reused her own code or not. Hence, the reported license violations may actually be cases of self-copying.

Conclusion validity threats concern the relation between the treatment and the outcome. In RQ3, we found some cases of "code migration" from an app (App_A) to Stack Overflow then to another app (App_B). The code snippet in App_B was created later than the one in Stack Overflow, which was created later than the one in App_A. However, another possibility could be that both Stack Overflow and App_B code snippets were reused from App_A. However, given the popularity of Stack Overflow, the chances that App_B copied from Stack Overflow are high. We strongly recommend that developers always provide a reference and the license of the original code in their derivative works posted on Stack Overflow. This will help prevent the community from turning Stack Overflow into a "code laundering platform".

External validity threats concern the possibility to generalize our results. The findings in this paper might not be generalized to other Q&A websites and–or other systems, since our datasets were limited to some selected Android apps and Stack Overflow code snippets. Although these datasets are very large and contain apps from different domains, future studies with other open-source systems and Q&A websites could help provide deeper insights on software license violation issues in Q&A websites. To help researchers replicate this work or conduct future works, we share our analytic scripts and data in Github: https://github.com/swatlab/stack_overflow.

VI. Related Work

In this section, we discuss related works that investigated Q&A websites and software licenses.

A. Question and Answer Websites

Q&A websites provide a platform for users to exchange knowledge. Gyöngyi et al. [16] investigated user behaviours in Yahoo! Answers, which is a Q&A website for general topics. The authors analyzed the popularity of top-level categories based on the number of questions and answers in each category. Adamic et al. [17] investigated knowledge sharing in Yahoo! Answers. They analyzed the characteristics of the website's users and their answers, and proposed models to predict whether a particular answer will be chosen as the best answer by the asker.

Since the introduction of Stack Overflow in 2008, a plethora of studies have focused on this Q&A website, designed for developers. Anderson et al. [18] proposed models to predict the long-term value of a question and its answers on Stack Overflow. They also proposed models to predict whether a question requires a better answer. Barua et al. [12] explored topics and trends on Stack Overflow. They observed the growth of mobile application development questions and the decline of questions about the .NET framework. They also observed that Git has surpassed SVN in terms of the impact of version control systems, and that Java is still an important programming language among developers. Vasilescu et al. [19] studied the interplay between Stack Overflow and the repository hosting website, Github. They observed that active Github committers ask fewer questions on Stack Overflow than others, and that

the questions on Stack Overflow tend to be associated with the social coding in Github. However, the author did not investigate whether developers copy and paste code between the two websites and whether they respect license restrictions. Ponzanelli et al. [20] implemented a tool, named *Prompter*, which automatically retrieves pertinent answers from Stack Overflow and provides them to developers in their IDE. Through a quantitative study, they showed that *Prompter* can identify the pertinent discussions if given a context in the IDE. Although this tool can provide useful suggestions, developers still need to pay attention to the licenses of the suggested code from Stack Overflow. In general, none of the previous studies has investigated whether developers respect software licenses when copying code from or to Q&A websites. Our study aims to fill this gap in the literature and raise the awareness of the software engineering community about potential unethical code reuse activities taking place on Q&A websites.

B. Software Licenses

Software licenses are legal instruments that govern the use or redistribution of software.

Vendome et al. [21] conducted a study on license usage and changes in 16,221 Java projects hosted on Github. They found that only 0.9% of commit messages mention software licenses. Also, they found no discussion related to licenses in the issue reports. The authors speculated that developers are too shy to document license changes. In a follow up study [9], the authors indicated that developers do not necessarily know the consequences of using a specific license into their code. They conducted a survey to understand when and why developers adopt and change licenses and observed that developers have difficulties dealing with license terms (*e.g.*, incompatible licenses). They also observed that developers often change licenses toward more permissive licenses to facilitate the reuse of their products in commercial systems. In a recent study [4], we investigated license violations in 857 Android apps using the license detection tool Ninka [22], and found 399 apps with license inconsistencies (*i.e.*, files that share similar code but have different licenses) and 17 apps with license violations.

Sojer and al. [23] investigated unethical code reuse from Internet-accessible sources through a survey of 869 professional software developers. They reported that developers who perform unethical code reuse have limited knowledge on license terms and do not understand the associated legal risks. The paper suggests that software organizations warn developers about the negative consequences of unethical code reuse, provide an environment that encourages compliance with laws, and avoid excessive time pressure. Although this paper studied inappropriate code reuse from the Internet, the authors did not attempt to quantify the occurrence of unethical code reuse in real open-source projects or Q&A websites.

VII. Conclusion

The question and Answer (Q&A) website, Stack Overflow, provides a platform for programmers to share expertise and exchange ideas. It allows users to reuse its content under certain restrictions. In this paper, we examine whether developers reuse code from Stack Overflow to Android apps and whether they share code from the Android apps to Stack Overflow. We found 232 Android code snippets that are exact clones of code snippets posted on Stack Overflow. These code snippets are distributed in 135 files from 62 different apps. We investigated the licenses of these 232 code snippets and observed potential cases of license violations in 60 apps. We also found 1,226 Stack Overflow posts that potentially reused code from Android apps, and 1,219 of these posts do not respect the original apps' license. In total, we detected 1,279 cases of potential license violations. Code snippets reused from Stack Overflow tend to stay in the apps for a long time. We also found 126 code snippets that seem to have migrated from one app to Stack Overflow and then from Stack Overflow to another app. In 12 of the code snippets, the file containing the code snippet in the first app and the file containing the code snippet in the second app used different software licenses. These findings suggest that developers do not pay enough attention to copyright terms when reusing code from Stack Overflow or sharing code on Stack Overflow. We hope that this paper will raise the awareness of the software community about potential unethical code reuse activities taking place on Q&A websites like Stack Overflow.

Acknowledgment

This work is partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and by Fonds de Recherche du Québec – Nature et Technologies (FRQNT). We gratefully thank the developers who participated in our survey.

References

[1] "9 lines of code that Google allegedly stole from Oracle's Java," https://fossbytes.com/9-lines-of-code-that-google-stole-from-oracle-java-android/, 2016, online; Accessed October 17th, 2016.

[2] "Creative Commons Attribution-ShareAlike 3.0 Unported License," https://creativecommons.org/licenses/by-sa/3.0/legalcode, 2016, online; Accessed September 20th, 2016.

[3] M. Sojer and J. Henkel, "License risks from ad hoc reuse of code from the internet," *Communications of the ACM*, vol. 54, no. 12, pp. 74–81, 2011.

[4] O. Mlouki, F. Khomh, and G. Antoniol, "On the detection of licenses violations in the Android ecosystem," in *Proceedings of the 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER 2016)*, vol. 1. IEEE, 2016, pp. 382–392.

[5] "Do I have to worry about copyright issues for code posted on Stack Overflow?" http://meta.stackexchange.com/questions/12527/do-i-have-to-worry-about-copyright-issues-for-code-posted-on-stack-overflow, 2016, online; Accessed Octobre 6th, 2016.

[6] J. Svajlenko and C. K. Roy, "Evaluating modern clone detection tools." in *Proceedings of the 30th International Conference on Software Maintenance and Evolution (ICSME 2014)*. IEEE, 2014, pp. 321–330.

[7] J. R. Cordy and C. K. Roy, "The NiCad clone detector," in *Proceedings of the 19th International Conference on Program Comprehension (ICPC 2011)*. IEEE, 2011, pp. 219–220.

[8] "Licenses & Standards," https://opensource.org/licenses, 2016, online; Accessed September 20th, 2016.

[9] C. Vendome, M. Linares-Vásquez, G. Bavota, M. Di Penta, D. M. German, and D. Poshyvanyk, "When and why developers adopt and change software licenses," in *Proceedings of the International Conference on Software Maintenance and Evolution (ICSME 2015)*. IEEE, 2015, pp. 31–40.

[10] "Summary of the Creative Commons Attribution-ShareAlike 3.0 Unported License," https://creativecommons.org/licenses/by-sa/3.0/, 2016, online; Accessed September 20th, 2016.
[11] C. K. Roy, J. R. Cordy, and R. Koschke, "Comparison and evaluation of code clone detection techniques and tools: A qualitative approach," *Science of Computer Programming*, vol. 74, no. 7, pp. 470–495, 2009.
[12] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? An analysis of topics and trends in Stack Overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.
[13] "Compatible Licenses," https://developer.android.com/training/implementing-navigation/nav-drawer.html, 2016, online; Accessed September 20th, 2016.
[14] M. Rieger, S. Ducasse, and M. Lanza, "Insights into system-wide code duplication," in *Proceedings of the 11th Working Conference on Reverse Engineering (WCRE 2014)*. IEEE, 2004, pp. 100–109.
[15] R. K. Yin, *Case Study Research: Design and Methods - Third Edition*, 3rd ed. SAGE Publications, 2002.
[16] Z. Gyöngyi, G. Koutrika, J. Pedersen, and H. Garcia-Molina, "Questioning Yahoo! Answers," in *Proceedings of the 1st Workshop on Question Answering on the Web*, 2008.
[17] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman, "Knowledge sharing and Yahoo answers: everyone knows something," in *Proceedings of the 17th International Conference on World Wide Web*. ACM, 2008, pp. 665–674.
[18] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Discovering value from community activity on focused question answering sites: a case study of stack overflow," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 850–858.
[19] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *Proceedings of the International Conference on Social Computing (SocialCom)*. IEEE, 2013, pp. 188–195.
[20] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza, "Mining stackoverflow to turn the ide into a self-confident programming prompter," in *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR 2014)*. ACM, 2014, pp. 102–111.
[21] C. Vendome, M. Linares-Vásquez, G. Bavota, M. Di Penta, D. German, and D. Poshyvanyk, "License usage and changes: a large-scale study of java projects on github," in *Proceedings of the 23rd International Conference on Program Comprehension (ICPC 2015)*. IEEE, 2015, pp. 218–228.
[22] D. M. German, Y. Manabe, and K. Inoue, "A sentence-matching method for automatic license identification of source code files," in *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering (ASE 2010)*. ACM, 2010, pp. 437–446.
[23] M. Sojer, O. Alexy, S. Kleinknecht, and J. Henkel, "Understanding the drivers of unethical programming behavior: The inappropriate reuse of internet-accessible code," *Journal of Management Information Systems*, vol. 31, no. 3, pp. 287–325, 2014.