What Do Practitioners Ask about Code Clone? A Preliminary Investigation of Stack Overflow

Eunjong Choi *, Norihiro Yoshida †, Raula Gaikovina Kula *, Katsuro Inoue *
* Graduate School of Information Science and Technology, Osaka University, Japan
{ejchoi, raula-k, inoue}@ist.osaka-u.ac.jp
† Graduate School of Information Science, Nagoya University, Japan
yoshida@ertl.jp

***Abstract*—We present a preliminary investigation of Stack Overflow to reveal practitioner's interests about code clones. We then discuss possible future directions of research on code clones.**

I. Introduction

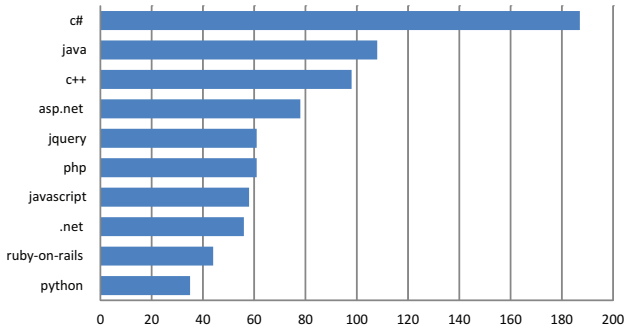
In recent decades, the clone research community has provided numerous techniques for the detection and analysis of code clones in source code [1]. However, very little is known in regards to true practitioner's needs on the detection and analysis of code clones. To provide a positive impact into industrial and OSS developments, the clone research community should be aware of what practitioners are interested in about code clones. Recently, several studies have investigated practitioner's interest by mining Q&As in Stack Overflow (SO) [2], [3], [4]. For example, Pinto and Kamei analyzed Q&As in *SO* to discover practitioner's needs for refactoring tools [4]. Their analysis found that practitioners do not often rely on the existing refactoring tools.

In this paper, we analyzed *SO* for investigating practitioner's needs for clone detection and analysis. The goal of this investigation is to find out whether code clone techniques and tools have met the requirements of programmers or contributed development techniques.

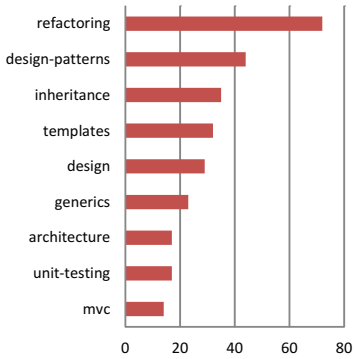
II. The Data that We Analyzed

This paper analyzed data provided by the MSR 2013 challenge [5]. The data contains the dump for the *SO* website. It is comprised of data related to questions, answers, the users that have created those data, and other information from July 30, 2008 to July 31, 2012.

To filter-out unrelated data, at first, we chose keywords related to the code clones. The chosen keywords are *code clone*, *code cloning*, *code redundancy*, *code duplicate*, *code duplication*, *duplicate code*, and *duplicated code*. Then, we selected questions that had at least one of above-mentioned keywords in titles, or in bodies, or in tags. The goal of this study is to reveal problems that practitioners tackle about code clones, therefore we only selected questions because they provide clues about what kind of difficulties practitioners have with code clones. As a result, 1,654 questions were selected. Finally, two of the authors manually validated the selected questions to filter out false positives. False positives included unrelated or duplicated logs or data. Consequently, 925 questions with topics related to code clones were found.



(a) Tags related to programming language



(b) Tags related to development techniques

Fig. 1. Tags related to programming language and development techniques

III. Investigation Results

This section details the investigation results and provides answers to two Research Questions (RQs).

RQ1: What Kinds of Programming Languages/Techniques were Appeared in Questions on Clones?

Methodology : To answer this RQ, we manually analyzed tags which are appeared in questions in *SO*. In details, after a total of 846 tags were identified from the questions, we grouped together related tags. For example, 'wicket' and 'wicket-1.5' tags were grouped together as a 'wicket' tag. Then, we categorized these tags to *Tags related to programming language* and *Tags related to development techniques*. With this process, tags which are unrelated to programming languages and development techniques were excluded. One

TABLE I. THE AVERAGE REPUTATIONS OF PEOPLE WHO ASKED AND ANSWERED POSTS ON CLONES, AND GENERAL

	Reputations(Clone)	Reputations(General)
Asked	2690	263
Answered	1395	410

example of the excluded tags is a 'code-duplication' tag which is appeared 142 times.

Finding : Figure 1 represents the numbers of tags related to programming languages and development techniques. As shown in figure 1(a), the most frequently appearing tags related to programming language are object-oriented programming languages such as 'c♯' [1], 'java', and 'c++' [2] tags. These tags on object-oriented programming languages are followed by tags on web programming languages. They are 'asp.net'[3], 'jquery'[4], 'php', 'javascript', 'ruby-on-rails' [5]. Moreover, tags on dynamic programming languages such as 'javascript', and 'python' also frequently appear. Figure 1(b) represents the numbers of tags related to development programming language and techniques. As shown this figure, tags related to clone management frequently appear. For example, the 'refactoring' tag involves questions about clone refactoring. Moreover, 'design-patterns' was tagged in questions related to the technique of clone refactoring using specific design-pattern such as the *factory* pattern. Tags such as 'inheritance', 'templates', and 'generics', which are related to clone refactoring also frequently appear.

Answer to RQ1 : The most frequently tags in questions are about web programming languages. Moreover, tags about clone management also frequently appear.

RQ2: Were Questions on Clones Asked by Trusted Practitioners?

Methodology : To answer this RQ, we compared the average reputations of practitioners who asked and did not ask about clones.

Finding : The results show that questions related to clone clones were asked by practitioners with a relatively higher reputation. (Table I). In *SO*, the reputation of each user is increased when other users vote on his/her questions or answers. Additionally, reputation is increased by an accepted answer [6]. The investigation result suggests that questions on code clones were mainly asked by practitioners who are trusted by the others in *SO*. Note that questions about clones were mainly answered by practitioners who have higher reputations compared to the others (Table I).

Answer to RQ2 : Practitioners who asked about clones are trusted by the others in *SO*.

[1]It contains 'c♯', 'c♯-3.0', and 'c♯-4.0' tags.

[2]It contains 'c++', 'c++11', and 'c++-faq' tags.

[3]It contains 'asp.net', 'asp.net-mvc', 'asp.net-mvc-2', 'asp.net-mvc-3', and 'asp.net-mvc-4' tags.

[4]It contains 'jquery', 'jquery-cycle', 'jquery-isotope', 'jquery-mobile', 'jquery-plugins', 'jquery-selectors', and 'jquery-templates' tags.

[5]It contains 'ruby-on-rails', 'ruby-on-rails-2', 'ruby-on-rails-3', and 'ruby-on-rails-3.2' tags.

[6]http://stackoverflow.com/help/whats-reputation

IV. DISCUSSION

According to our investigation, code clones written in scripting languages are more frequently discussed in *SO* (e.g., Javascript, PHP, and Python). Main reason considered is that web applications developed by scripting languages may include a significant amount of clones. An investigation by Rajapakse and Jarzabek [6] also suggests a similar result. The clone research community mainly has focused on clones in large-scale source code written in Java and C/C++ and only a few techniques have been proposed on the detection of clones written in scripting languages [7]. Therefore, the clone research community should focus not only Java and C/C++ but also scripting languages.

Also, our investigation tells us that the most discussed techniques were refactoring related. The result recognizes the need for basic support of development of tools for clone refactoring [8], [9]. Discussions on clone refactoring, template, generics, and inheritance were often discussed as techniques for avoiding and merging clones. Design patterns are also discussed with design and architecture for removing clones [10]. Practitioners in *SO* discussed about not only the removal of existing clones but also prevention of clones during class/architecture design phase. The clone research community should focus not only maintenance of existing clones but also prevention of clones during the design phase. *SO* practitioners that asked questions on clones have relatively higher reputations. This indicates that relatively higher-level members of *SO* are interested in clones.

Only a few questions were discussed on the usage of clone detection tools, and several questions were created by practitioners who looked for clone detection tools. The result indicates that more promotion of clone related research is strongly needed.

As future work, we plan to investigate the latest posts of *SO* using unsupervised learning to categorize questions related to code clones to achieve generality of our findings.

REFERENCES

[1] C. K. Roy and J. R. Cordy, "A survey on software clone detection research," School of Computing, Queen's University, Tech. Rep. 2007-541, 2007.

[2] 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.

[3] M. Allamanis and C. Sutton, "Why, when, and what: analyzing Stack Overflow questions by topic, type, and code," in *Proc. of MSR*, 2013, pp. 53–56.

[4] G. Pinto and F. K. Kamei, "What programmers say about refactoring tools?: An empirical investigation of Stack Overflow," in *Proc. of WRT*, 2013, pp. 33–36.

[5] A. Bacchelli, "Mining challenge 2013: Stack Overflow," in *MSR*, 2013.

[6] D. C. Rajapakse and S. Jarzabek, "An investigation of cloning in web applications," in *Proc. of ICWE*, 2005, pp. 252–262.

[7] P. Bulychev and M. Minea, "An evaluation of duplicate code detection using anti-unification," in *Proc. of IWSC*, 2009.

[8] G. P. Krishnan and N. Tsantalis, "Unification and refactoring of clones," in *Proc. of IEEE CSMR-WCRE*, 2014, pp. 104–113.

[9] R. Tairas and J. Gray, "Increasing clone maintenance support by unifying clone detection and refactoring activities," *Information & Software Technology*, vol. 54, no. 12, pp. 1297–1307, 2012.

[10] J. Kerievsky, *Refactoring to Patterns*. Addison Wesley, 2004.