



The Influence of Code Retrieval from the Web on Programmer's Skills, Methodologies, and Coding Behaviors

Alfarooq O.M. Mohammed^{1*}, Ziad A. Abdelnabi² and Abdalmunam Abdalla¹

¹Department of Computer Science, Omar Al-Mukhtar University, Al Bayda, Libya

²Department of Computer Science, The Libyan Academy of Graduate Studies, Benghazi Libya

Received: 18 March 2021/ Accepted: 15 June 2021

Doi: <https://doi.org/10.54172/mjsc.v36i2.66>

Abstract: The development of software projects consists of several stages, such as analysis and design. It also requires a set of skills that the software developer can use to work on the project, such as specifying the requirements and writing code. Developers usually search for source code on the internet for remix and reuse in software production. This paper aims to investigate the influence and effect of code retrieved from the web on programmers' views, decisions, and skills. A questionnaire instrument was designed and distributed to programmers for their feedback. As a result, we were able to address some points and achieved a better understanding of the interaction between programmers and the code from the web, especially the code from programming forums such as Stack Over Flow.

Keywords: Code Reuse; Source Code Search; Code Influence; Code Remix

INTRODUCTION

With a rising number of code forums and open source websites, the internet is considered to be a large repository of information regarding the source code of different programming languages available for every software developer. The amount of source code on the internet has increased dramatically because of the distribution of open-source repositories. Similarly, the term open source has become very commonplace in the past few years (Gabriel, 2005). The most important benefit of having open-source repositories, is that they form a rich source for software projects on the web that can be reused and redeveloped (Szyperski, 2004). For instance, GitHub.com, forge.net, and freshmeat.net host many software projects that support programmer participation and sharing knowledge. An alternative way of finding source code is through the use of general-purpose search engines such as Google and Yahoo as programmers are accustomed to such tools and their ability to get the required

documents from the web (Gallardo-Valencia, 2013).

In the past, searching for code has been investigated comprehensively, and researchers suggested many approaches to refine the performance of code search (Haiduc, 2013; Lee Martie, 2017; Haoran Niu, 2017; Mukund Raghathan, 2016). According to research in software engineering (Janice Singer, 1997), searching the web is the most common activity for software engineers, and program understanding, software reuse, debugging was mentioned as the top prompting for source code searching in that research. In another related study on searching for source code (S. E. Sim, 1998), The search purposes mentioned frequently by the software developers were; code reuse, debugging, program understanding, feature addition, and effect analysis. Software developers also use code search engines (e.g., Krugle, Github native search) to get their code snippets from open source repositories using general natural language queries (Lopes, 2012),

*Corresponding Author: Alfarooq O.M. Mohammed Alfarooq.mohammed@omu.edu.ly, Department of Computer Science, Omar Al-Mukhtar University, Elbeida, Libya.

but many of them are now obsolete, and programmers prefer to use a general-purpose search engine to look for code (Graham, 2016; Susan Elliott Sim, 2011; Stolee, 2014).

Another study investigated the performance of searching with Google and whether it varies for code versus non-code connected search (Md. Masudur Rahman, 2018). Another research has proposed a new technique which could help developers in their search for code by identifying related and particular API classes from a programming task site, written as a natural language query, and then reformulated for better source code search (Roy, 2018). According to Szyperski (2004), software reuse is a prevalent motivation for source code searching on the internet for programmers. In 2018, a study proposed a neural code searching technique for source code retrieval that uses natural language to search for code (Sachdev, 2018). Other research presented a new code retrieval framework that combines both ordering and semantic relationships and demonstrates significantly improved retrieval precision between two known datasets (Akbar, 2019).

Although code retrieval has been an active research topic in recent years, less focus was on the impact of retrieved code on programmers' skills and coding behavior. In this study, the focus is on studying and analyzing the effect and influence of code retrieval from the web on programmers' views, decisions, and abilities. Through reviewing previous studies, it was noted that they were mainly investigating the methods and techniques of searching for code and the effectiveness of online discussion forums. We believe that there are always factors that influence the success or failure of software development projects. We also believe that retrieving code from the web to be deployed in software projects has an effect on the developers.

MATERIALS AND METHODS

Code searching on the internet (CSOI) is an important method for sharing best coding

practices. In recent years, software properties have led to programming complexity. Programmers' goal is to implement software design by investigating on similar coding practices which best fit their needs. The aim of the current research is to investigate the issues behind implementing CSOI; the goal is to inspect the factors reflecting on CSOI as a measurable searching process from the viewpoint of the programmer.

We assume that CSOI is a measurable process. When performing CSOI, programmers' top priority is to gain similar code or better coding techniques. If the required source code from CSOI is obtained, we can say that the CSOI results are successful. However, if the programmer was unable to attain the required results from CSOI, then the CSOI results are unsuccessful, and CSOI must be repeated with different factors and priorities.

In recent years, the increased usage of the internet has been accompanied by an equal increase in online surveys that have been used to investigate human behavior over the internet (Ritter., 2011). To achieve the goal of this study, an online survey with 26 ended questions was designed to be distributed to a wide range of programmers who often search for source code on the internet. The population was all programmers with at least six years of programming experience and conducted code searching on the internet (CSOI). The survey was available for about two months in 2019 to collect the required feedbacks. Invitations for participants were shared and sent via email addresses to university academic programmers and private business sectors programmers. This category of candidates was selected because they represent the programming communities in the country and have common interests regarding searching for source code. Thus, potentially better responses could be obtained.

The study sample consisted of 37 respondents. The study first started by cleaning the data set and preparing data for analysis. The

preprocessing step produced 37 cases ready for analysis. The collected data indicated that most of the respondents are C Sharp and Visual Basic programmers with more than seven years of experience.

Analysis: The collected data was analyzed by both quantitative and qualitative techniques. The questionnaire was tested for reliability using alpha Cronbach method (Taber, 2018). The test showed good reliability of 82.8%. The questions were based on a five-point Likert scale with the following possible choices: 1 – strongly disagree, 2- disagree, 3- neutral, 4- agree, and 5- strongly agree. The mean and standard deviation of each question were computed to determine the outcome of each question.

Examining the data individually, we discovered a high level of agreement in our analysis. Then, we merged our analysis and refined the results for clarity of presentation.

THREATS TO VALIDITY

The fundamental deficiency of this study is generalizability, i.e., the number of respondents is insufficiently representative of the population. This is a major problem with empirical research in software engineering as there is an unreliable pattern of population characteristics so that the representativeness of a sample can be estimated.

This research is no exception. Moreover, participants were selected from mailing lists and programming groups. That is why there is no attempt to quantify the propagation of some types of behavior.

Instead, we are searching for a diversity of search behaviors and patterns, which is suitable for a discovery study. The objectives of this study are to find answers to the following four research questions:

Q1. To what extent "Code retrieval from the web" improves programmer's knowledge and skills?

Q2. What is the impact of "Code retrieval from the web" on limiting programmer's views when trying to solve coding problems and bottlenecks?

Q3. What is the best criterion by which to measure the quality of "Code retrieval from the web"?

Q4. What are the advantages and disadvantages of using "Code retrieval from the web"?

RESULTS AND DISCUSSION

Most respondents agreed that they memorize most of the functions in the language they use. They also agreed that they use online programming forums more frequently than the help page files provided with a certain language when trying to solve programming problems. This is similar to the findings of the study of Janice Singer (1997), which reported that searching the web is the most common activity for software engineers. They also reported that code retrieval neither made them change the programming language they use nor forced them to change the version of the programming language itself. It also did not make them adapt to new programming concepts.

Respondents think that retrieved code is scalable and the majority are attempting to understand the retrieved code before using it. Most of them make libraries from the retrieved code for later use. They neither agree nor disagree about having difficulties embedding retrieved code into their projects. They also were neutral about the fact that retrieved code satisfies verification, security, and control procedures of their systems.

Most of them stated that they only use the first solution they find online to their problems without further search for other solutions that could be more flexible and more scalable. Moreover, they claim that code retrieval did not make them dependent on others to solve their

programming problems. The majority of the respondents to this study stated that they do not participate in solving problems posted on online programming forums and they usually find solutions to their problems without posting them. Some respondents claim that there are some projects in which they did not use retrieved code and sometimes they did not find solutions to their problems or even solutions similar to them.

Regarding the first question of the study, the majority of respondents stated that they use retrieved code to understand how a certain

function works rather than directly copy and paste the code into their projects. This is similar to the study of (S. E. Sim, 1998) that showed that program understanding is among the search purposes mentioned frequently by software developers. They also believe that they acquire more knowledge and learn from the code they retrieve. For the questions set in the questionnaire related to the first question of the study as shown in table 1, the average was 3.99, which means there is an agreement that retrieved code has some degree of improvement on the programmer's skills, coding practice, and problem-solving ability.

Table (1): Result of question one of the study

	Strongly disagree	disagree	neutral	agree	Strongly agree	Mean	STDev	Outcome
Q 1	0	0	3	14	20	4.46	0.65	Strongly agree
Q 2	2	4	4	10	17	3.97	1.24	Agree
Q 3	4	5	3	17	8	3.54	1.28	Agree

The second question of the study investigates the impact of retrieved code on limiting programmers' views when solving programming problems. The average of this dimension of the study, as shown in table 2, was 2.89, which indicates that there is no significant impact of retrieved code on limiting programmers' views in programming. Interestingly, the majority of respondents seemed to use the first solution they found online, which may indicate the programmer's reliance on that particular solution with the

possibility of a better solution not being explored.

This on the one hand, could lead to producing less quality code and may be a sign of the limitation of the programmer's skills. On the other hand, considering the advancement in code search engines, the first solution produced by those engines, to some degree, is always a good solution as many users had given it a higher rank. Hence, these first-hit solutions are considered to contribute to better code and improve programmer's skills.

Table (2): Result of question two of the study

	Strongly disagree	disagree	neutral	agree	Strongly agree	Mean	STDev	Outcome
Q 1	15	6	5	8	3	2.41	1.42	Disagree
Q 2	7	4	4	15	7	3.30	1.41	Neutral
Q 3	9	3	6	10	9	3.19	1.52	Neutral
Q 4	10	10	2	13	2	2.65	1.36	Neutral

The third question of the study inspects the best criterion by which to measure the quality of code retrieval from the web. The study revealed that clarity, scalability, security, and ease of

implementation are the most used factors by programmers for the assessment of retrieved code quality.

The last dimension of the study investigates the advantages and disadvantages of code retrieving practice. Here, the study showed the following advantages of retrieving code from the web. Code retrieval from the web provides an easy way to obtain code that can be embedded in various projects and solve numerous problems without the need to read long pages of help files. It prompts the programmer towards learning, understanding, and sometimes adopting new software development concepts and programming methodologies. Code retrieval makes programmers familiar with programming sites and online programming forums and increases their participation in finding solutions to programming problems.

The study also pointed out the following disadvantages of code retrieval from the web: Sometimes solutions found online do not fit directly into one's project and need to be customized, rewritten, or even changed from one language to another to fit into the project.

In some cases, programmers may find themselves frequently adopting new different programming concepts just because they can easily find code that follows those concepts, which may take some time and reduce productivity.

CONCLUSION

This paper has discussed the influence of code retrieval on programmers' skills from a programmer's point of view. The results obtained after careful analysis of the responses that were collected via the designated questionnaire suggested that code retrieval may improve programmers' skills. The study also revealed that there is no notable impact of code retrieval practice on limiting programmers' views when solving programming problems and developing software projects. The research has pointed out some factors, namely: clarity, scalability, security, and ease of implementation, that programmers think are the best criteria to measure the quality of retrieved code. Finally, the paper has shown

some advantages and disadvantages of code retrieval from the web.

Further future work can be done on this topic, including studying exactly which particular programming skills can be improved by code retrieval from the web. Also, future work may focus on investigating which methods and practices programmers use when dealing with code that does not fit directly into their projects.

Furthermore, further investigation can be done on the reasons why respondents consider retrieved code to be scalable and the majority of them are attempting to well understand the retrieved code before using it. Moreover, future studies could focus on investigating the reasons that make programmers less proactive in sharing solutions to problems on programming forums.

REFERENCES

- Akbar, S. A. (2019). SCOR: Source Code Retrieval with Semantics and Order. *IEEE Press*, 1–12.
- Gabriel, R. G. (2005). *Innovation Happens Elsewhere: Open Source as Business Strategy*. San Francisco, CA: Morgan Kaufmann Publishers.
- Gallardo-Valencia, S. E. (2013). *Finding Source Code on the Web for Remix and Reuse*. New York Heidelberg Dordrecht London: Springer.
- Graham, M. H. (2016). Software search is not a science, even among scientists. *CoRR*, *abs/1605.02265*.
- Haiduc, S. a. (2013). Automatic Query Reformulations for Text Retrieval in Software Engineering. *Proceedings of the 2013 International Conference on Software Engineering*. *IEEE Press*, 842--851.
- Haoran Niu, I. K. (2017). Learning to Rank Code Examples for Code Search

- Engines. *Empirical Software Engineering*, 22(1), 259--291.
- Janice Singer, T. L. (1997). An examination of software engineering work practices. *In Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative Research*, 21.
- Lee Martie, A. v. (2017). Understanding the Impact of Support for Iteration on Code Search. *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ACM., 774--785.
- Lopes, S. K. (2012). Analyzing and mining a code search engine usage log. *EMSE*, (4-5(17), 424--466.
- Rahman, M. M., Barson, J., Paul, S., Kayani, J., Lois, F. A., Quezada, S. F., Parnin, C., Stolee, K. T., & Ray, B. (2018). Evaluating how developers use general-purpose web-search for code retrieval. *Proceedings of the 15th International Conference on Mining Software Repositories*,
- Mukund Raghothaman, Y. W. (2016). SWIM: Synthesizing What I Mean: Code Search and Idiomatic Snippet Synthesis. *In Proceedings of the 38th International Conference on Software Engineering*. ACM, 357--367.
- Ritter., V. M. (2011). *Conducting Online Surveys*. Thousand Oaks, CA.: SAGE Publications.
- Roy, M. M. (2018). Effective Reformulation of Query for Code Search Using Crowdsourced Knowledge and Extra-Large Data Analytics. *IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 473-484.
- Sim, S. E., Clarke, C. L., & Holt, R. C. (1998). Archetypal source code searches: A survey of software developers and maintainers. *Proceedings. 6th International Workshop on Program Comprehension. IWPC'98 (Cat. No. 98TB100242)*,
- Sachdev, S. a. (2018). Retrieval on Source Code: A Neural Code Search. *Association for Computing Machinery*, 31--41.
- Stolee, K. T. (2014). Solving the Search for Source Code. *ACM Trans. Softw. Eng. Methodol*, 23, 3.
- Susan Elliott Sim, M. U. (2011). How Well Do Search Engines Support Code Retrieval on the Web? *ACM Trans. Softw. Eng. Methodol.*, 21, 25.
- Szyperski, D. S. (2004). Guest editors' introduction: How is open source affecting software development? *IEEE Software*, 21(1), 28-33.
- Taber, K.S. (2018). The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education. *Res Sci Educ* 48, 1273-1296.

تأثير استخلاص الشفرة البرمجية من الإنترنت على مهارات وطرق وسلوك المبرمج

الفاروق عمر مفتاح^{1*}، زياد عبد النبي² وعبد المنعم سعد فضل الله³

¹ قسم الحاسوب، جامعة عمر المختار، البيضاء، ليبيا

² قسم الحاسوب الأكاديمية الليبية، بنغازي، ليبيا

تاريخ الاستلام: 18 مارس 2021 / تاريخ القبول: 15 يونيو 2021

<https://doi.org/10.54172/mjsc.v36i2.66>:Doi

المستخلص: في تطوير مشاريع البرمجيات، يتكون المشروع البرمجي من عدة مراحل على سبيل المثال التحليل، والتصميم. وهو يتطلب أيضاً مجموعة من المهارات التي يمكن لمطور البرامج استخدامها للعمل في المشروع على سبيل المثال، تحديد المتطلبات، وكتابة التعليمات البرمجية. يبحث المطورون عادةً عن التعليمات البرمجية على الإنترنت لإعادة المزج، وإعادة الاستخدام في مشاريع البرامج الخاصة بهم. تهدف هذه الورقة إلى التحقق من تأثير الشفرة المستخلصة من الويب على آراء المبرمجين، وقراراتهم، ومهاراتهم. ندرس هذا التأثير من وجهة نظر المبرمج باستخدام استبانة تم إعدادها وتوزيعها على مجموعة من المبرمجين للحصول على آرائهم، وتعليقاتهم نتيجة لذلك تم إلقاء الضوء على بعض النقاط، وتم تحقيق فهم أفضل للتفاعل بين المبرمجين، والكود المستخلص من الويب، وخاصة الكود المستخلص من منتديات البرمجة مثل: (Stack Overflow)

الكلمات المفتاحية: إعادة استخدام الشفرة البرمجية، استخلاص الشفرة البرمجية من الإنترنت، تأثير الشفرة البرمجية، مزج الشفرة البرمجية.