LETTER *Special Section on Empirical Software Engineering*

Analyzing Web Search Strategy of Software Developers to Modify Source Codes**

Keitaro NAKASAI[†*a)], ***Nonmember***, **Masateru TSUNODA**[††], ***and*** **Kenichi MATSUMOTO**[†], ***Members***

SUMMARY Software developers often use a web search engine to improve work efficiency. However, web search strategies (e.g., frequently changing web search keywords) may be different for each developer. In this study, we attempted to define a better web search strategy. Although many previous studies analyzed web search behavior in programming, they did not provide guidelines for web search strategies. To suggest guidelines for web search strategies, we asked 10 subjects four questions about programming which they had to solve, and analyzed their behavior. In the analysis, we focused on the subjects' task time and the web search metrics defined by us. Based on our experiment, to enhance the effectiveness of the search, we suggest (1) that one should not go through the next search result pages, (2) the number of keywords in queries should be suppressed, and (3) previously used keywords must be avoided when creating a new query.

key words: *web search strategy, software development, subjective experiment*

1. Introduction

Considerable useful programming information is available on the World Wide Web. For example, many official references to programming languages and Q&A websites (e.g., StackOverflow) are available. Software developers often use web search engines, such as Google, to obtain useful information that increases work efficiency [7].

Web search strategies (such as "changing search phrases frequently") might be different for each developer. If the search strategy is ill-suited, it could take a long time to find a web page that contains useful information, resulting in a decrease in work efficiency of programming. To avoid such a situation, we analyzed the search strategies of developers to establish guidelines for preferable web searches.

Many previous studies analyzed web search behavior in programming [2], [3], [6], [7], [9], [11], mostly analyzing how web searches are employed (e.g., frequency and search target). For example, Sadowski et al. [7] analyzed how often and what developers search for on the Web. However, these studies did not provide guidelines for web search strategies.

Manuscript received February 26, 2021.
Manuscript revised July 12, 2021.
Manuscript publicized October 29, 2021.

[†]The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.
[††]The author is with Kindai University, Higashiosaka-shi, 577–8502 Japan.
[*]Presently, with NIT, Kagoshima College.

[**]This work is an extended study of K. Nakasai et al., "Web Search Behaviors for Software Development," In Proc. of International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2016), pp.125–128, 2016.

a) E-mail: nakasai@kagoshima.kosen-ac.jp

DOI: 10.1587/transinf.2021MPL0004

2. Web Search Metrics

We defined five web search metrics to quantitatively analyze the web search strategies of software developers. To define the metrics, we focused on the selection of search keywords (input into the search engine) and the understanding of search results (output from the search engine). This is because they are considered important for analyzing web search behavior.

Result pages Per Viewed pages (RPV)

$$\mathrm{RPV} = r \,/\, v \tag{1}$$

The web search engine creates a web page, which includes a list of websites related to the query. We refer to the created web page that includes the list as *a search result page*. When Google is used as the search engine, a search result page contains 10 web page URLs by default. The value r is the number of times the search result page is shown during programming (task). The value v is the number of web pages that are visited during programming (the value includes r).

For example, if a developer accesses a search result page during programming, $r = 1$. The developer also visits 10 web pages (i.e., a search result page and nine ordinal web pages that are not search result pages) during programming; therefore, $v = 10$. In this example, RPV = 0.1.

The typical reaction to the search result output can be classified into two groups. One is to judge whether each page is useful by reading only the title and summary of the web page included in the search results, and the other is not only to read the search results, but also to access the web pages included in the search results and to read the content. When the RPV is high, developers tend to do the former, and when it is low, they tend to do the latter. The aforementioned numerical example falls in the second group. An example of the first case is that a developer accesses nine search result pages and an ordinal web page. Therefore, $r = 9$, $v = 10$, and RPV = 0.9.

Unique Queries Per Result pages (QPR)

$$\mathrm{QPR} = q \,/\, r \tag{2}$$

A query is defined as a string of words that a developer inputs into a web search engine. For example, "class interface" and "null pointer exceptions" are queries. The value q is the number of unique queries used during programming. For example, if the queries are "class interface,"

"null pointer exception," and "class interface," the value is 2. The definition of r is the same as that of RPV.

As explained, a Google search result page includes 10 web links, the destinations of which are related to the query by default. To check more than 10 links, a developer should go to the next search result page. Similarly, to check more than 20 links, the developer should go to a search result page after the next page. When accessing the search result pages sequentially, the denominator of the QPR increases but the numerator remains the same.

Therefore, when the QPR is small, developers tend to go to the next result page. For instance, when the query is "class interface," and the developer browses through three search result pages, (i.e., he/she checked 21–30 web links), $q = 1$ and $r = 3$. As a result, the value of the QPR becomes 0.33. In contrast, a large QPR implies that the developer does not frequent the next result pages.

The value of QPR also decreases when the developer cannot produce a new query and uses the same query again. In this case, the denominator also increases while the numerator remains the same. This possibility was considered in the analysis.

All key Words Per unique Queries (WPQ)

$$\mathrm{WPQ} = w \,/\, q \tag{3}$$

In the above equation, w is the number of keywords included in unique queries. For example, if the query is "class interface," the value is 2. The value q is defined in QPR. That is, WPQ represents the average number of keywords in each query. Google's search guidelines suggest that it is preferable to include fewer keywords in queries. To analyze the validity of the guidelines in programming, we used WPQ. When WPQ is large, it implies that the developer used many keywords in queries.

Result pages Per Task time (RPT)

$$\mathrm{RPT} = r \,/\, t \tag{4}$$

The value t is the time required for the developer to complete the programming (i.e., task). The value *of* r is defined in the RPV. For example, if a developer accesses 10 search result pages in 10 minutes, the RPT is 1. A higher RPT indicates that the developer gains knowledge from the Web (or is reading search result pages), instead of programming or reading non-search result pages.

Unique key words Per all key Words (UPW)

$$\mathrm{UPW} = u \,/\, w \tag{5}$$

The value u refers to the number of unique keywords used during programming. For example, used queries are "class interface" and "class abstract," the value u is 3 and the value w is 4. A higher UPW indicates that the developer uses the same keywords when creating a new query.

3. Experiment

3.1 Overview

In the experiment, we asked the subjects four questions about programming, and they tried to solve them. The subjects were ten students majoring in information science. One of them was a master course student, one a second-year undergraduate, and the remaining were third-year students (ages of all the students were approximately 20). Although the number of subjects was fairly small, some studies used 10 subjects [4], [5]. The subjects used Google Chrome as a web browser, Google as a search engine, and Eclipse as an editor. Some metrics are based on the number of accessed web pages, which, in turn, is based on the history of the title of the web browser; the history is collected using software (Key Logger Free Edition) [10].

Although the subjects were not professional software developers, the study [8] suggests that the difference between students and professionals is little. Thus, students can substitute practitioners. We believe that the results of the analysis would not be significantly different if practitioners were used as subjects. However, experiments using professionals as subjects will be conducted in our future studies.

3.2 Questions about Programming

The questions about programming were linked to Java because all the subjects studied Java at their universities. In the questions, the source code that should be modified was given in advance. Each source code had a defect, and the subjects were asked to search through the Web and understand the cause of the defect and how to fix it. The source code was modified based on the knowledge obtained. After the experiment, the subjects were asked whether they knew the solution before the experiment.

Question one: In a given source code, the floating-point operation raises an error. In the question, we asked how to fix the problem using a Java library [1]. To solve this problem, subjects must find an appropriate library.

Question two: A given source code uses an array list as the data structure to store the data. Subjects should modify the program using an associative array. The question does not directly indicate the use of an associative array. However, an associative array should be used to fulfill the function described in the question.

Question three: In a given source code, data is received from a website. Character codes are shown, although the expected result is an html document format. To solve this problem, subjects must find an I/O library of Java.

Question four: In a given source code, an appropriate error message is not shown. To solve the problem, subjects need to understand the exception hierarchy of Java.

Definition of correct answer: If the subject answered the current question correctly, the next question was displayed. That is, the next question was shown only when the subject's answer to the previous question was correct.

Definition of task time: The task time started when a question was displayed to the subjects and ended when JUnit confirmed that the program was modified correctly. Notably, the task time includes not only the web searching time, but also the coding time of the program (i.e., the

time to modify the program based on the web search results). However, the coding time was considered short and did not differ much among the subjects. Therefore, this would not affect the analysis. This is because it is easy to modify source codes in the questions if an appropriate library is found. Task time is discussed in detail in Sect. 5.2.

Definition of incorrect answer: No time limit was set for answering the questions. However, if the task time for a question exceeded 20 min, the question could be skipped. The skipping of a question was considered as an incorrect answer and the skipped question could not be answered again (e.g., when a subjects started to answer Question 2, it could not return to Question 1).

3.3 Research Questions

To clarify the goal of the analysis, we formulated the following research questions. In the questions, "beter" means "effective to shorten task time."

- **RQ1**: To judge the utility of the web pages shown in the search results, is it better to access the pages in addition to reading their summaries included in the results?
- **RQ2**: Is it better to go through next search result pages?
- **RQ3**: Is it better to suppress the number of keywords in queries?
- **RQ4**: Is it better to gain more knowledge from the Web (i.e., showing more result pages) in a short time?
- **RQ5**: Is it better to avoid previously used keywords when creating new queries?

4. Results

Analyzed data: Ten subjects answered four questions, and as a result, 40 data points were collected. In seven cases out of the 40 data points, subjects knew the library (solution) to be applied in advance. The number of correct answers was seven for Question 1, 8 for Question 2, and 6 each for Questions 3 and 4. Three subjects answered all the questions correctly, and one subject answered all the questions incorrectly.

When we excluded seven cases where subjects knew the solution in advance, the number of available data points was 33. Additionally, when we excluded 13 cases of incorrect answers (i.e., skipped questions), 20 data points were used for the analysis. Because the number of data points is not very large, we also analyzed the 33 data points which included incorrect answers, as a reference.

Relationships to task time: We assumed that shorter task time means higher work efficiency and analyzed the relationship between task time and Web search metrics. Spearman's rank correlation coefficient was used to analyze the relationships to avoid the influence of outliers. In psychology, if the absolute value of the correlation coefficient is larger than 0.2, the relationship is regarded as a weak correlation. In the subject experiment of software engineering, the condition was similar to that of the psychology experiment. Therefore, we focused on correlations with an absolute value larger than 0.2.

The correlation coefficients and p-values between the task time and web search metrics are shown in Table 1. The absolute values of the correlation for all metrics were larger than 0.2 for the 20 correct answers. Of the 33 data points, including incorrect answers, the absolute values were also larger than 0.2, except for RPV.

4.1 Analysis to Answer RQ1

RPV: Although RPV was positively correlated with task time, we cannot answer "yes" to RQ1. The 20 data points were divided into two groups (longer task time and shorter task time) according to the median of the task time. The median number of accesses to destination pages was 16 in the longer task group and 7 in the shorter task group. That is, the number of accesses to the pages was large in the first group (RPV was not positively related to time).

Table 1 Correlation coefficients between task time and web search metrics

		RPV	QPR	WPQ	RPT	UPW
Include incorrect answers (33 data points)	ρ	0.038	-0.291	0.245	0.266	-0.557
	p-value	0.833	0.101	0.169	0.135	0.001
Only correct answers (20 data points)	ρ	0.341	-0.255	0.325	0.305	-0.621
	p-value	0.142	0.278	0.162	0.191	0.003

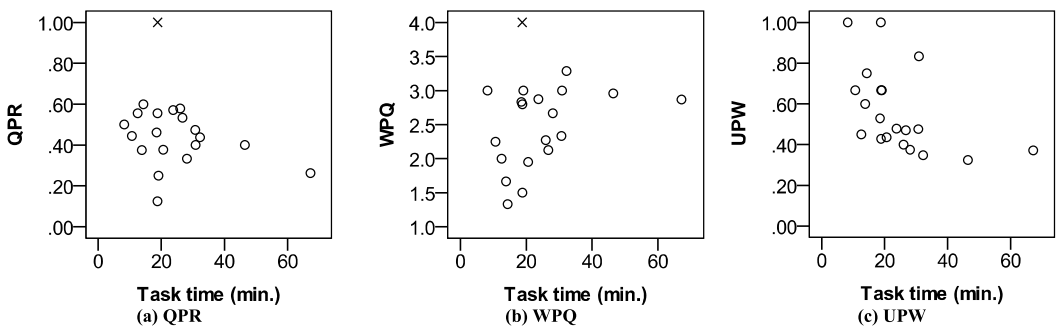

Fig. 1 Relationships between web search metrics and task time

4.2 Analysis to Answer RQ2

QPR: Figure 1 (a) shows a scatter plot of the task time for 20 data points of the correct answers. In the figure, the data point denoted by "X" is considerably different from the other data points. When we exclude the outlier, the correlation coefficient becomes −0.254 (p-value = 0.296), which is almost the same as when the outlier is not removed.

We cannot explicitly observe a monotonic decreasing trend in the figure. However, when the task time was shorter (less than approximately 30 min), there were more data points whose QPR was larger (i.e., larger than approximately 0.4). In contrast, when the task time was longer, there were more data points with a smaller QPR. Therefore, a negative correlation is observed in the figure.

In Fig. 1 (a), we focus on data points, the task times of which were shorter than the median (i.e., data points on the left side of the graph), and those with task times longer than the median (i.e., data points on the right side). If the longer-time subjects often used the same query again, the numerator of the QPR did not increase but the denominator increased. As a result, the tendency of the QPR is identical to that of the graph. In this case, the QPR indicates how often a query is reused but does not indicate how often it goes to the next result pages.

However, the numerator's average and median were 7.95 and 7, respectively, for the shorter-time subjects, and 12.62 and 14, respectively, for the longer-time subjects. That is, the number of queries used by the longer-time subjects increased. Therefore, when the QPR is small, the subjects tend to not use the same query repeatedly but go through the next result pages.

According to the results, when the QPR is small, that is, when a developer goes to the next result pages more often, the task time tends to be longer. Therefore, the answer to RQ2 is "no."

4.3 Analysis to Answer RQ3

WPQ: The scatter plot of the task time for 20 correct answers is shown in Fig. 1 (b). In the figure, the data point denoted by "X" is considerably different from the other data points. When the outlier was excluded, the correlation coefficient became 0.415 (p-value = 0.077), and the correlation became stronger when the outlier was not excluded. In the figure, it can be observed that the minimum and maximum WPQ of data points whose task time is approximately 10 min, are lower than that of data points whose task time is approximately 30 min.

In the former case, the range of the WPQ was approximately 1.5–2.5, and approximately 3.0, in the latter case. In some cases, the task time was equal to the medium (approximately 20 min) and the WPS was around 3.0, as shown in the figure. Therefore, although it is better to use fewer than three words in each query, this is not a necessity. All subjects used Japanese for the web search engine. Therefore, the above range of the WPQ could vary when English or other languages are used for the web search.

From the results, it can be observed that when the WPQ is large, that is, when the developer includes several keywords in the query, the task time tends to be longer. Therefore, the answer to RQ3 is "yes."

4.4 Analysis to Answer RQ4

RPT: RPT was positively correlated with task time. This indicates that when the task time increases, web searches per task time also increase. This simply indicates that the task is stuck (i.e., the number of searches increased because an appropriate web page was not found). Therefore, the answer to RQ4 is "no."

4.5 Analysis to Answer RQ5

UPW: The p-value was smaller than 0.05. That is, the correlation was statistically significant at 5%. The scatter plot of the task time for 20 correct answers is shown in Fig. 1 (c). From the figure, we can see that when the UPW is small, the task time tends to be long.

Therefore, when the UPW is high, that is, the new query does not include the keywords used before, the task time tends to be short. That is, the answer to RQ5 is "yes." To choose new keywords, it might be better to use synonyms of the keywords used previously.

5. Discussion

5.1 Analysis Focusing on Correctness of Answers

To understand the web search metrics from another perspective, we focused on the relationship between the correctness of answers and the web search metrics. To analyze the relationship, we created a dummy variable that denotes whether the answer is correct or incorrect; the value is 0 in the former case and 1 in the latter cases. Same as Table 1, we calculated Spearman's rank correlation coefficient between the dummy variable and web search metrics.

The correlation coefficients using 33 data points are shown in the top row of Table 2. We only focused on the QPR, WPQ, and UPW because they are related to work efficiency, as explained in Sect. 4. Although the absolute values are smaller than those in Table 1, the positive and negative values of each metric are identical to those in Table 1. We speculated that the absolute values in the top row of Table 2 decreased owing to data points with correct answers but longer task time.

Table 2 Correlation coefficients between the correctness of answers and web search metrics

Dummy variable		RPV	QPR	WPQ	RPT	UPW
correct answers (0) or not (1)	ρ	0.046	-0.235	0.020	0.065	-0.111
	p-value	0.801	0.189	0.914	0.719	0.539
correct and time $\leq$ 66 percentile (0) or not (1)	ρ	0.045	-0.319	0.262	0.172	-0.309
	p-value	0.805	0.071	0.141	0.339	0.080

To suppress the influence of such data points, we also set the dummy variable to 1 when the answer of the data point was correct but its task time was longer than 66 percentile of the time. Intuitively, we also regarded the data points as incorrect answers when the task time was longer. As a result, the value of the dummy variable became 1 for 4 of the 20 correct answers.

Under the above conditions, as shown in the bottom row of Table 2, the correlation coefficients of the QPR, WPQ, and UPW were similar to those in Table 1. Therefore, even when we consider the correctness of answers, the proper web search strategies suggested in Sect. 4 still apply.

5.2 Relationship between Coding and Task Time

In the analysis, it would be better to remove the coding time from the task completion time. However, coding and searching can be frequently switched during the tasks. For instance, a subject finds an undesired web page and modifies the program incorrectly. In this case, the program does not pass the unit test prepared by us, and the subject tries again to find desired web pages. Therefore, it is difficult to automatically distinguish the search time from the task time.

For 20 correct answers, the average number of web pages visited per minute was 1.32, and the median was 1.04. The average number of webpages visited during the task was 33.5. In the case of 33 data points, including incorrect answers, the values were 1.45, 1.19, and 61.6 respectively. The results suggest that subjects spent most of their task time searching. This is because the tasks typically require subjects to find proper libraries but not to formulate appropriate algorithms. Therefore, the tasks can be completed easily by using proper libraries but cannot be completed without them. For example, given the source code for question one is the following:

```
double answer = 2.00 - 1.10
```

To set the value of the answer as 0.90, the BigDecimal class should be used as follows:

```
BigDecimal answer = new BigDecimal("2.00")
   .subtract(new BigDecimal("1.10"))
```

Although the modification is easy, the value of the answer cannot be set as 0.90 without the BigDecimal class. Therefore, the coding time was considered short and did not differ much among the subjects.

5.3 Threats to Validity

Internal validity: The tasks in the experiment required the subjects to find proper libraries. However, the details of the tasks were different, and the difference could be a confounding factor in the analysis. This issue could be easily solved by classifying the data per question. However, the classification reduces the data points considerably, making their analysis difficult. Therefore, we did not classify the data; however, we should be aware of the possibility of confounding in the tasks before utilizing the analysis results.

External validity: The experimental results were only observed in the tasks to find proper libraries. Developers could also use web search engines to identify the cause of system failure, such as runtime errors. A future research undertaking is to clarify a better web search strategy for other types of tasks.

6. Conclusion

We analyzed the web search strategy of developers in programming. We defined web search metrics to analyze the strategies. In the experiment, subjects answered questions about programming (the usage of Java libraries), gaining knowledge about the questions from the Web. In the analysis, we focused on the relationship between web search metrics and task time to solve these problems. The analysis suggests that search strategies are preferable for reducing time.

- It is recommended not to go through next search result pages much.
- It is better to suppress the number of keywords in queries.
- When creating a new query, it is better to avoid previously used keywords.

In actual software development, developers make use of searches on the Web for various reasons. The above search strategies are expected to be effective, especially when developers use libraries for programming. Note that, owing to the small size of the experiment, the obtained results are preliminary. We plan to conduct further research to validate these strategies in a future work.

Acknowledgments

This research was partially supported by the Japan Society for the Promotion of Science (JSPS) [Grants-in-Aid for Scientific Research (C) and (S) (No.21K11840 and No.20H05706).

References

[1] J. Bloch and N. Gafterj, Java Puzzlers: Traps, Pitfalls, and Corner Cases, Addison-Wesley Professional, 2005.
[2] S. Bajracharya and C. Lopes, “Analyzing and mining a code search engine usage log,” Empirical Software Engineering, vol.17, no.4, pp.424–466, 2012.
[3] R. Gallardo-Valencia and S. Sim, “Information used and perceived usefulness in evaluating web source code search results,” Proc. CHI ’11 Extended Abstracts on Human Factors in Computing Systems, pp.2323–2328, 2011.
[4] T. Mizuno, S. Nomura, A. Nozawa, H. Asano, and H. Ide, “Evaluation of the effect of intermittent mental work-load by nasal skin temperature,” IEICE Trans. Inf. & Syst., vol.J93-D, no.4, pp.535–543, 2010 (in Japanese).
[5] T. Nakagawa, Y. Kamei, H. Uwano, A. Monden, and K. Matsumoto, “On measuring the difficulty of program comprehension based on

cerebral blood flow," Computer Software, vol.31, no.3, pp.270–276, 2014 (in Japanese).

[6] M. Rahman, J. Barson, S. Paul, J. Kayani, F. Lois, S. Quezada, C. Parnin, K. Stolee, and B. Ray, "Evaluating how developers use general-purpose web-search for code retrieval," Proc. International Conference on Mining Software Repositories (MSR), pp.465–475, 2018.

[7] C. Sadowski, K. Stolee, and S. Elbaum, "How developers search for code: a case study," Proc. Joint Meeting on Foundations of Software Engineering (ESEC/FSE), pp.191–201, 2015.

[8] I. Salman, A. Misirli, and N. Juristo, "Are students representatives of professionals in software engineering experiments?," Proc. International Conf. on Software Engineering (ICSE), pp.666–676, 2015.

[9] S. Sim, M. Umarji, S. Ratanotayanon, and C. Lopes, "How Well Do Search Engines Support Code Retrieval on the Web?," ACM Transactions on software Engineering and Methodology, vol.21, no.1, article 4, 2011.

[10] Sword, Key Logger Free Edition, http://keylog.web.fc2.com/keyfree/keyfree.html

[11] X. Xia, L. Bao, D. Lo, P. Kochhar, A. Hassan, and Z. Xing, "What do developers search for on the web?," Empirical Software Engineering, vol.22, no.6, pp.3149–3185, 2018.