An Eye Tracking Study on camelCase and under_score Identifier Styles

Bonita Sharif and Jonathan I. Maletic
Department of Computer Science
Kent State University
Kent, Ohio 44242
bsimoes@cs.kent.edu and jmaletic@cs.kent.edu

***Abstract*— An empirical study to determine if identifier-naming conventions (i.e., camelCase and under_score) affect code comprehension is presented. An eye tracker is used to capture quantitative data from human subjects during an experiment. The intent of this study is to replicate a previous study published at ICPC 2009 (Binkley et al.) that used a timed response test method to acquire data. The use of eye-tracking equipment gives additional insight and overcomes some limitations of traditional data gathering techniques. Similarities and differences between the two studies are discussed. One main difference is that subjects were trained mainly in the underscore style and were all programmers. While results indicate no difference in accuracy between the two styles, subjects recognize identifiers in the underscore style more quickly.**

Keywords-identifier styles; eye-tracking study; code readability

I. INTRODUCTION

The comprehension of identifier names in programs is at the core of program understanding. Identifier names are often key beacons to program plans that support higher-level mental models of understanding. According to Deißenböck et al. [11] identifiers make up approximately 70% of source code. If a certain identifier naming style significantly increases the speed of code comprehension, this could significantly impact overall program understanding.

Currently we have two main styles for identifiers, namely camel-case (e.g., studentGrade) and underscore (e.g., student_grade). In the work presented here, we study the comprehensibility of these two styles and attempt to determine if one is significantly better than the other. Our goal is to add to the basic understanding of how we comprehend identifiers so that coding standards [23] can reflect the most efficient techniques.

Early programming languages such as Basic, COBOL, Fortran, Pascal, and Ada were case insensitive and programmers were encouraged to use underscores to separate compound identifier names. With the advent of case-sensitive languages such as C, C++, Python, and Java, the trend has been to use camel-case style identifiers. This may, in part, be due to the fact that it is a bit easier and faster to type a camel-case identifier than it is an underscore identifier. The position of the underscore on the keyboard and the number and combination of keystrokes required plays a role in typing speed. However, does the ease of writing identifiers affect the accuracy of code readability and maintainability?

To address this topic, Binkley at al. [4] conducted a study with 135 subjects consisting of programmers and non-programmers to determine which identifier style was faster and more accurate. They hypothesized that identifier style affects the speed and accuracy of software maintenance. The subjects (who had programming experience) were mostly trained in the camel-case style. The study used an online game-like interface to gather timed responses from the subjects. Their findings show that camel-cased identifiers lead to higher accuracy among all subjects, and those trained in the camel-case style, were able to recognize camel-cased identifiers faster. However, with respect to all subjects, camel-cased identifiers took 13.5% longer than underscored identifiers ($p\text{-}value<0.0001$).

Here, we attempt to replicate Binkley et al.'s [4] experiment using an eye tracker to gather eye gaze data during the experiment. In our study, only programmers (experts and novices) are used as subjects. All of our subjects had experience with both styles and their preferences of style was approximately split even among the group. In addition, most of the subjects were historically trained in the underscore style. The main task of the study remains the same as Binkley's, which is to pick the correct identifier from a group of four closely related, although different, identifier names. Results from eye tracking studies done in the domain of cognitive psychology [12, 21] on reading un-spaced text imply that camel-cased identifiers should be more difficult to read compared to underscored identifiers. We believe that replicating the experiment using an eye tracker will add to the empirical evidence as to which style is faster and more accurate for comprehension.

The paper is organized as follows. Section II describes the research questions the paper addresses. The design of the experiment is presented in Section III. Results are analyzed in Section IV followed by a discussion. Section VI outlines the threats to validity. Related work is presented in Section VII, followed by conclusions and future work.

II. RESEARCH QUESTIONS

The goal of this study is to analyze human subjects' eye-gaze data while they perform the tasks of correctly detecting an identifier from a group of four closely related identifiers. Although this task is relatively simple as pointed out by Binkley et al. [4], it gives insight into the readability aspect of identifier styles. With the data generated from an eye tracker we know the exact location of where the subject is looking, the duration of the subject's gaze at a particular location, and movement between different locations on the screen. These measures lead to a fine-grained analysis thus generating more refined conclusions. Although there are many eye tracking studies related to evaluating user interfaces [3, 9, 10, 14, 19, 20], there are very few studies done by few researchers on how programmers read and comprehend source code [1, 2, 8, 25]. To bridge this gap, we conducted an eye-tracking replication of Binkley et al.'s [4] study since the topic lends itself well to eye tracking analysis.

The main research questions this paper addresses are:

- RQ1: Does identifier style affect the accuracy and time needed to read and detect correct identifiers?
- RQ2: Is the visual effort needed to read and detect correct identifiers the same for camel-case and underscore styles?

III. EXPERIMENTAL DESIGN

We describe the experiment based on the template given by Wohlin et al. [26]. The experiment *seeks to analyze* the effect identifier style has on searching for correct identifiers *for the purpose of* evaluating their usefulness in code readability and comprehension *with respect to* effectiveness (accuracy) and efficiency (time) *from the point of view of* the researcher *in the context of* students at Kent State University.

An overview of the experiment is given in Table I. The main factor being analyzed is the identifier style used. The dependent variables are discussed in Section III.D. We also examined secondary factors such as the effect experience has on the dependent variables. The experiment is conducted as a within-subjects design where all subjects are given both treatments of the main factor and a paired comparison between identifier styles is made between them. In Binkley et al.'s experiment, repeated measures were used however the exact details of the number of subjects in each group are not reported in the paper. Since our sample size is small (N=15), we wanted to gather more data points for each style and use a within-subjects comparison.

TABLE I. EXPERIMENT OVERVIEW

Goal	Study the effect identifier style has on code readability
Independent variable	Identifier style (Style) with two treatments: camel-case or underscore
Dependent variables	Correctness, Find Time, Visual Effort
Secondary factors	Reading Time, Experience, Phrase Length, Phrase Origin, Style Preference, Visual Effort on Reading Phrase
Design	Within-subjects

A. *Eye-tracking Apparatus*

The *Tobii 1750* eye tracker (www.tobii.com) is used for this study. It is a video-based remote eye tracker that uses two cameras to capture eye movements. The cameras are built into a 17 inch TFT-LCD hardware. The screen resolution was set to 1024 by 768. This eye tracker does not require the subject to wear any form of head gear, thereby emulating a subject's normal work environment. The frame rate (temporal resolution) at which sampling occurs is 50 Hz, latency is around 25-35 ms, and average accuracy is 0.5 degrees (approx. 15 pixels average error). The eye tracker compensates for head movement during the study i.e., the eyes do not have to be focused on the screen all the time.

The *ClearView* analysis software that comes with the eye tracker was set up as a double screen configuration. The first screen is used by the moderator to set up and run the study. The second screen is used by the study subjects to perform the tasks. This lets the moderator get real time feedback of the eye tracking quality during the task. The Tobii eye tracker records eye gaze and audio/video recordings of the entire study session. The eye gaze data include timestamps, gaze positions, eye positions, pupil size, and validity codes.

B. *Material and Stimuli*

The main objects of this study are a set of eight phrases (same as Binkley et al.'s study). The subject first reads a phrase and when they are done studying it, the next screen asks them to choose an identifier (from four choices) that exactly matches the phrase they just saw. Fig. 1 shows the *phrase stimulus* and the *question stimulus* for each task. There are eight such tasks. See Table II for the set of phrases used. Only one of the choices is correct, the rest are distracters that change the beginning, middle and end of the identifier. For detailed information about the identifier selection process and distracters used, we direct the reader to [4]. Unlike the Binkley's study, the clouds on the question stimuli do not move and the phrase is not shown on the question stimuli. Since the previous study does not indicate which style was used to generate identifiers for each phrase, we randomly assigned a style to each phrase within each phrase type.

Each of the identifier phrases is characterized by a length, origin, and style used. The length is the number of words in the phrase. Phrase origin determines whether or not the phrase is likely to be in source code. For example, *river bank* is a 2-word non-code phrase since the probability of finding it in source code is low. The reason for including non-code phrases in the original study was to determine if familiarity with a phrase had an effect on performance.

The presentation order of the questions is shown in the second column of Table II. The order was determined using Latin squares to avoid learning biases. During the analysis, we do a pair-wise comparison between the four pairs: Q1 and Q5, Q7 and Q3, Q4 and Q2, Q6 and Q8. Instead of testing each subject on the camel-case and underscore versions of the same identifier (causing learning effect), a different but similar identifier in the opposing style is used. It is important to note that corresponding underscore or camel-cased versions of each phrase in Table II, were not

used in this study i.e., the underscore style for *start time*, *river bank*, *extend alias table* and *movie theater ticket* and the camel-cased style for *full pathname*, *drive fast*, *get next path*, and *read bedtime story* were not used in this study.

C. Visual Effort and Areas of Interest

The idea behind eye tracking is that visual attention (focus on a particular location) triggers mental processes to comprehend or solve a given task [17]. Based on this correlation, we can study the cognitive behavior and effort involved in solving a task. Visual effort is denoted by the amount and duration of eye movements, in certain areas of the stimuli, needed to verbally state the correct answer.

We analyze our results only based on areas of interest and not on eye gaze data on the blank part of the screen. Two main types of eye gaze data are eye fixations and saccades. A *fixation* is the stabilization of the eye on an object of interest for a period of time, whereas *saccades* are quick jerky movements from one fixation to another. It has been determined that comprehension mainly takes place during fixations and not during saccades. The eye tracker was set to filter fixations within 20 pixels with a duration of at least 40ms. This is the standard setting recommended for reading for the Tobii 1750 eye tracker.

Visual effort is studied with respect to certain areas of interest (AOI) on the stimuli. These are presented below. For each *phrase stimulus*, we define two areas of interest.

- Reading Task: The task description shown on the top of the phrase stimulus in bold face font. It instructs the participant to study the phrase.
- Phrase: The phrase shown on the phrase stimulus i.e., *full pathname*, shown in Fig. 1. This area of interest is represented by the letter *P*.

For each *question stimulus*, six areas of interest are created.

- Entire stimulus: The task description and all four clouds. This area is represented by the letter *Q*.
- Question Task: The task description shown at the top of the question stimulus.
- Correct cloud: The cloud that correctly represents the phrase from the phrase stimulus.
- Distracter clouds: The three incorrect clouds representing distracters.

The areas of interest are represented as rectangles enclosing the task description, phrase and clouds and were constructed with a buffer zone of at least 50 pixels to accommodate for any small drifts of the eye tracker.

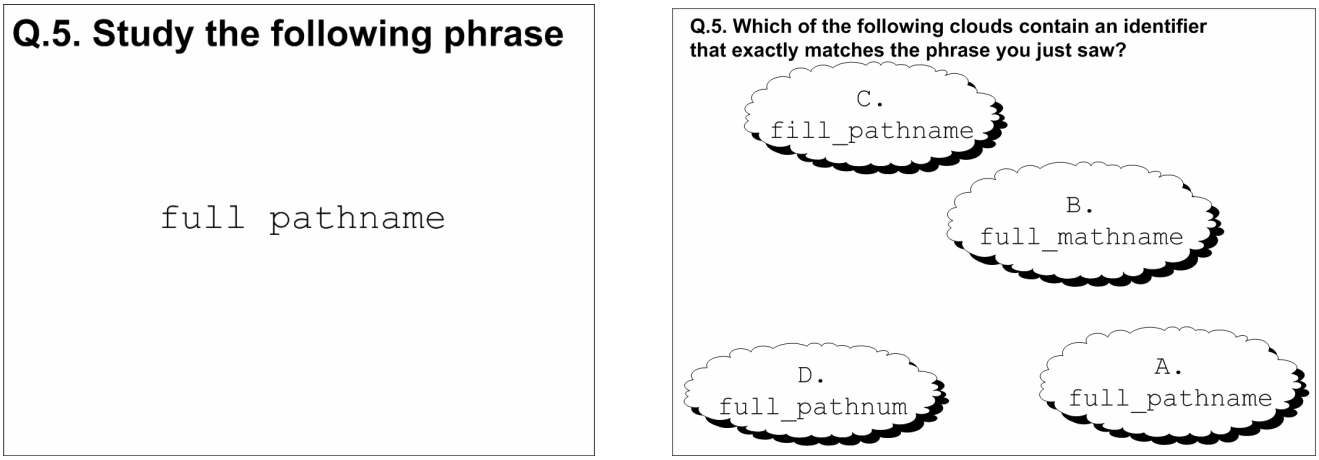

Figure 1. Phrase Stimulus (left) with the task description in bold and the phrase to be studied. Question Stimulus (right) with clouds to detect the correct identifier formed using the phrase presented on the Phrase Stimulus. The task description is shown at the top left corner of the screen.

TABLE II. PHRASES USED IN THE STUDY.

Phrase type	ID	Style	Phrase	Distracters Used (Begin, Middle, End)
2-word code	Q1	camelCase	start time	smart time, start mime, start tom
	Q5	under_score	full pathname	fill pathname, full mathname, full pathnum
2-word non-code	Q7	camelCase	river bank	riser bank, river tank, river ban
	Q3	under_score	drive fast	drove fast, drive last, drive fat
3-word code	Q4	camelCase	extend alias table	expand alias table, extend alist table, extend alias title
	Q2	under_score	get next path	got next path, get near path, get next push
3-word non-code	Q6	camelCase	movie theater ticket	mouse theater ticket, movie thunder ticket, movie theater ticker
	Q8	under_score	read bedtime story	raid bedtime story, read bedsore story, read bedtime store

D. Study Variables

The study consists of one independent variable, *identifier style*. The two values associated with this main factor are camel-case and underscore. The dependent or response variables are described next.

- Correctness: Denotes the accuracy of the answer verbally stated by a subject.
- Find Time FT(Q): Denotes the time taken by a subject to verbally state the correct answer. This is recorded in milliseconds.

Visual effort is determined using each of the following six individual measures. They are defined in terms of the areas of interest defined in the previous Section. The first three measures are based on eye fixations. A higher fixation count and fixation rate indicates more effort needed by subjects to solve the task.

- Fixation Count on Question Stimulus *FC(Q)*: The total number of eye fixations on all five areas of interest on the *question stimulus*. This refers to the entire stimulus.
- Fixation Rate on Correct Identifier *FR(correct)*: The total number of eye fixations on the correct identifier cloud with respect to all four clouds on the *question stimulus*.
- Fixation Rate on Distracters *FR(distracters)*: The total number of eye fixations on the distracter clouds with respect to all four clouds on the *question stimulus*.

The next three measures are based on eye fixation durations. The unit of measure is milliseconds. The more time spent analyzing the stimuli in search of an answer indicates more effort needed by subjects to solve the task.

- Average Fixation Duration on Question Stimulus: *AFD(Q)*: The average length of time of all fixations in all five areas of interest on the *question stimulus*.
- Average Fixation Duration on Correct Identifier: *AFD(correct)*: The average length of time of all fixations on the correct identifier cloud on the *question stimulus*.
- Average Fixation Duration on Distracters: *AFD(distracters)*: The average length of time of all fixations on the distracter clouds on the *question stimulus*.

Secondary variables are factors that might interact with the independent variable to have an effect on the dependent variables. These are described next.

- Phrase Length: The length of the phrase is determined by the number of words in the phrase. Phrases of length two and three are used.
- Phrase Origin: Determines whether or not the phrase is likely to be in source code. Possible values include *code* and *non-code*.
- Reading Time *RT(P)*: Denotes the time (ms) taken by a subject to study a phrase on the *phrase stimulus* before proceeding to the *question stimulus*.
- Experience: Indicates the level of expertise of the subjects. Two levels, experts and novices, were determined based on programming experience, number of years worked, and number of years in Computer Science. This variable combines the *Years Worked* variable and the *Training* variable used in the Binkley's study. The difference here is that we look for expertise within programmers rather than looking for differences between programmers and non-programmers.
- Style Preference: Denotes a subject's identifier style preference. Three values associated with this variable are camel-case, underscore, and no preference.

The next two variables are related to the visual effort needed to study a phrase on the *phrase stimulus*. It is measured using the following two variables.

- Fixation Count on the Phrase **FC(P)**: The total number of eye fixations on the phrase area of interest on the *phrase stimulus*. This does not include fixations on the task.
- Average Fixation Duration on the Phrase: **AFD(P)**: The average length of time of all fixations in the phrase AOI on the *phrase stimulus*.

The visual effort measures described above are summarized below.

$$FC(Q) = \sum_{a \in \{task,\ all\ clouds\}} f(a) \quad (1)$$

$$FR(correct) = \frac{\sum_{a \in \{correct\ cloud\}} f(a)}{\sum_{a \in \{correct\ cloud\} \cup \{distracter\ clouds\}} f(a)} \quad (2)$$

$$FR(distracters) = \frac{\sum_{a \in \{distracter\ clouds\}} f(a)}{\sum_{a \in \{correct\ cloud\} \cup \{distracter\ clouds\}} f(a)} \quad (3)$$

$$AFD(Q) = \frac{\sum_{a \in \{task,\ all\ clouds\}} g(a)}{\sum_{a \in \{task,\ all\ clouds\}} f(a)} \quad (4)$$

$$AFD(correct) = \frac{\sum_{a \in \{correct\ cloud\}} g(a)}{\sum_{a \in \{correct\ cloud\}} f(a)} \quad (5)$$

$$AFD(distracters) = \frac{\sum_{a \in \{distracter\ clouds\}} g(a)}{\sum_{a \in \{distracter\ clouds\}} f(a)} \quad (6)$$

$$FC(P) = \sum_{a \in \{phrase\}} f(a) \quad (7)$$

$$AFD(P) = \frac{\sum_{a \in \{phrase\}} g(a)}{\sum_{a \in \{phrase\}} f(a)} \quad (8)$$

where *f(a)* gives the fixation count and *g(a)* gives the total gaze time in an area of interest *a*.

In this study, we did not measure the amount of time spent on demographics (conducted at the end of the study) and the age of subjects. The questions on demographics were conducted verbally and in an interview-like setting.

This time varied greatly depending on the verbiage used. Since this was more open ended, we do not include it as a variable in our analysis.

E. Hypotheses

Based on the research questions posed in Section II, we generate six null hypotheses. See Table III. The alternative hypotheses do not assume directionality and simply state that the distribution is not same between identifier styles.

The first two hypotheses are the same as presented in Binkley et al.'s study. $H1_0$ seeks to determine if identifier style has an effect on correctness. In this case, correctness refers to the subject accurately stating the correct identifier built using the corresponding phrase. The second hypothesis ($H2_0$) seeks to determine if identifier style has an effect on the Find Time. In this case, Find Time refers to the time needed to verbally choose an answer from the *question stimulus*.

Hypotheses 3 ($H3_0$) and 4 ($H4_0$) are similar to the previous study using the *Experience* variable instead of *Training*. They seek to determine if experience interacts with identifier style to have an effect on Correctness and Find Time respectively. The last two hypotheses relate to eye-tracking measures defined in Section III.D. Hypothesis 5 ($H5_0$) seeks to determine if identifier style has an effect on the visual effort necessary to solve the task of recognizing the correct identifier. Finally, hypothesis 6 ($H6_0$) investigates the interaction effect of the secondary variable Experience with identifier style on visual effort.

F. Participants

The study participants were fifteen volunteers from the Department of Computer Science at Kent State University. There were seven undergraduates in their second year of study, eight graduate students, and two faculty members. Subjects were historically trained mostly in the underscore identifier style and were all programmers. All subjects had normal vision. Some wore contact or corrective lenses. The subjects were not aware of the experiment's hypotheses.

The following demographic data was collected for each subject after the study was completed: years in the CS program, years of experience in programming, years of working experience and identifier style preference. Based on this information, two groups of expert and novice programmers were determined. The expert programmers were termed as experts due to their involvement in industry and active participation in open source projects.

TABLE III. NULL HYPOTHESES

$H1_0$	There is no significant difference in Correctness between the camel-case and underscore identifier style (Style)
$H2_0$	There is no significant difference in Find Time between the camel-case and underscore identifier style (Style)
$H3_0$	The effect of Style on Correctness is independent of Experience
$H4_0$	The effect of Style on Find Time is independent of Experience
$H5_0$	There is no significant difference in Visual Effort between the camel-case and underscore identifier style (Style)
$H6_0$	The effect of Style on Visual Effort is independent of Experience

G. Instrumentation

The study was conducted in a dedicated room for the eye-tracking equipment. The subjects were seated approximately 60 cm away from the screen. An informed consent form was read and signed. The next step was calibrating the eye tracker for the subject. A five-point calibration was used (taking approximately one minute). During calibration, a subject focused their eyes on five points that appear on the screen (four for each corner, 1 for the center). The background color of the calibration was set to white since this was the background of the stimuli used in the study.

The first screen displayed instructions on what the task was. Next, two sample questions: one camel-case and one underscore, illustrating how to answer the questions were presented. After the subject understood the goal of the exercise, the actual study began. For each of the eight tasks, the *phrase stimulus* was presented first followed by the *question stimulus* (See Fig. 1). After the subjects were done studying the *phrase stimulus*, they said "next" to proceed to the *question stimulus*. The moderator controlled the movement through the tasks to avoid any unnecessary timing delays between subjects. The subjects were asked to verbally state the answer using the letter (i.e., A, B, C, or D) placed on top of the identifier in the cloud. After the eight identifier recognition tasks, an object location task was administered (See Section IV.C). The experiment took 13 minutes on average. Finally, after all the tasks were completed, the moderator debriefed each participant to gather some demographic data in an interview-like manner. This concluded the experiment.

IV. EXPERIMENTAL RESULTS

In order to facilitate comparison to the Binkley study, a linear mixed-effects regression model is fit to the Find Time dependent variable. In addition, since our study was within-subjects, the non-parametric paired Wilcoxon test is used to determine significance for the visual effort measures. Effect sizes using Cohen's d are noted to make results comparable with future studies on the topic.

A. Correctness and Find Time

Only one subject answered one question (Q4) incorrectly. This question used the phrase *extend alias table* in camel-case style. The subject chose *extendAliasTitle* (distracter at the end of the identifier). This is in line with the distracter analysis done in [4] that reports mistakes in camel casing occur more frequently when a change occurs at the end of the phrase. In this case, we cannot reject the null hypothesis ($H1_0$) and the subsequent related hypothesis ($H3_0$). This implies that there is no significant difference in accurately recognizing an identifier in either style. No further statistical analysis is needed here. In the Binkley et al. study, the odds of being correct are 51.5% higher for camel-cased identifiers, using a simple logistic GLMM (p-value=0.0250).

We now investigate the second hypothesis ($H2_0$), which examines the effect of identifier style on the speed of finding

the correct identifier. Fig. 2 presents the distribution of the Find Time dependent variable.

The data for Find Time was found to be normally distributed using the Shapiro-Wilk normality test. Similar to the original study, a simple linear mixed-model (at 95% confidence) is first fit to the data, where only Style is considered as an explanatory variable. In the simple model, the parameter estimate for Style is statistically significant (Style *p-value*=0.037, Intercept *p-value*<0.0001). On average, camel-cased identifiers took 932ms (20%) longer than underscored identifiers. In this case, we can reject the null hypothesis ($H2_0$). Binkley et al. report a p-value < 0.0001, where camel-cased identifiers take 13.5% longer.

A second model was fit to the data and included the secondary variable Experience as an explanatory variable in addition to Style, to determine if it interacts with Style to have an effect on Find Time ($H4_0$). In this model, Style is still statistically significant (*p-value*=0.035). See Table IV. However, Experience does not significantly interact with Style to have an effect on Find Time (*p-value* =0.472). In this case, we can't reject the null hypothesis ($H4_0$).

Even though the result is not statistically significant, we can make some observations about the findings. The interaction plot is given in Fig. 3. There is a larger time difference between experts and novices with respect to underscored identifiers (364ms), whereas the difference is less for camel-cased identifiers (279ms). Another observation is that the difference in time between identifiers styles within experts is much less (630ms) compared to the difference for the novice category (1275ms). This implies that experts are not affected as much as novices by the identifier style used.

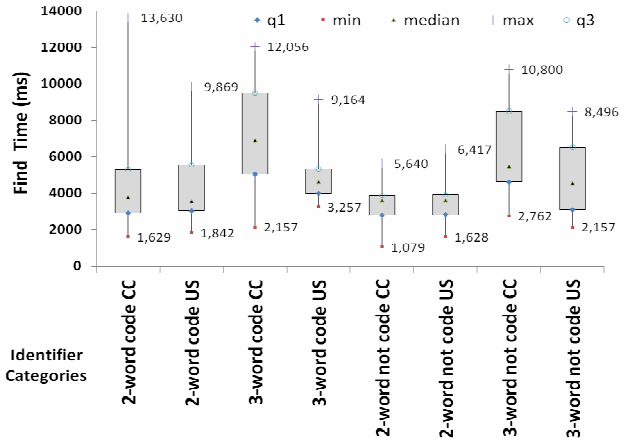

Figure 2. Descriptive Statistics for Find Time FT(Q) in each category of identifier style: camel-case (CC) and underscore (US)

TABLE IV. MODEL PARAMETERS FOR INTERACTION BETWEEN EXPERIENCE AND STYLE (α=0.05)

Variable	Value	Standard Error	t	*p*-value
Intercept	5307.53	431.423	12.302	**<0.0001**
Style	-953.17	445.572	-2.139	**0.035**
Experience	-42.50	445.572	-0.095	0.924
Style * Experience	-644.65	893.131	0.722	0.472

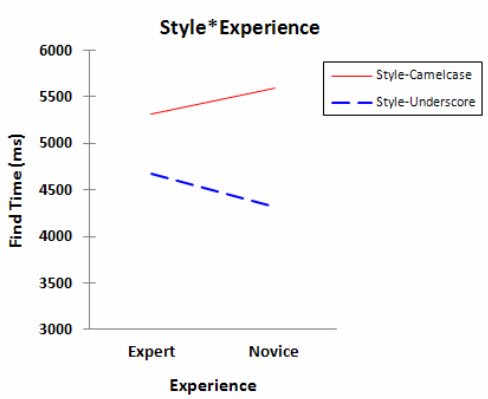

Figure 3. Interaction between subjects' experience and identifier style.

Finally, we fit a third complex model to the Find Time response variable to mimic the analysis of the original study. To determine if there are any other confounding variables, this complex model includes all secondary variables common to the original study discussed in Section III.D. This included Style, Style Preference, Experience, Phrase Origin, Read Time, and Phrase Length including all interactions between Style and each variable. The interaction between Phrase Origin and Experience is also included.

After backward elimination of non-significant terms, the model is presented in Table V. This model confirms the significance of Style. The model reports Phrase Length to be significant (p-value<0.0001). Phrase Length also significantly interacts with Style to have an effect on Find Time (p-value=0.048). Identifiers consisting of phrases of length three take 45% longer than phrases of length two. We also see that phrases of length two for camel-cased identifiers take approximately the same time as corresponding underscore identifiers however, for phrases of length three camel-cased identifiers take 36% longer than corresponding underscore identifiers.

Results of the Wilcoxon test on Find Time, FT(Q), are presented next (Row 1 of Table VI). Considering all camel-cased and underscored identifiers (global measure), a significant difference (*p-value*=0.01) in Find Time is noted, with camel-cased identifiers taking longer overall. The effect size is moderate (Cohen's d=0.57), which is considered to be practically significant. Taking a closer look, we find this significance only in the 3-word phrases. No significance is detected within the 2-word code/non-code identifiers, although camel-cased identifiers take longer on average. We also note that the read time, RT(P) of phrases for each style follow the same distribution (not shown here).

TABLE V. COMPLEX MODEL PARAMETERS FOR FIND TIME

Variable	Value	Standard Error	t	*p*-value
Intercept	-1358.9	1596.49	-0.851	0.39634
Style	-931.6	442.78	-2.104	**0.038**
Phrase Length	2718.8	626.19	4.342	**< 0.0001**
Style * Phrase Length	-1768.3	885.57	-1.997	**0.048**

B. *Visual Effort*

Visual Effort is measured by the six measures defined in Section III.D. Three measures relate to the number of fixations and the rest relate to the time involved in those fixations. Results of the Wilcoxon test for each of these measures are given in Table VI.

With respect to the fixation count FC(Q), and fixation rate: FR(correct) and FR(distracters), no significant difference was found between identifier styles with respect to the entire data set grouped by identifier style (*p-values*=0.213, 0.599, 0.599: rows 2 through 4 in Table VI.

For FC(Q) (total number of fixations on the question stimulus, Q), grouping identifiers by phrase length does give a significant improvement favoring underscore identifiers for both 2-word (*p-value*=0.029) and 3-word (*p-value*=0.033) identifiers, suggestive of a larger number of fixations for camel-cased identifiers. On average, there are six more fixations on 3-word camel-cased identifiers (i.e., *extendAliasTable*, *movieTheaterTicket*) compared to 3-word underscored identifiers (i.e., *get_next_path*, *read_bedtime_story*). The 2-word identifiers differ by only two fixations.

Breaking down the categories even further shows significance only for 2-word non-code identifiers (*p-value*=0.004), with 3-word code identifiers approaching significance (*p-value* = 0.057). The rate of fixations on correct and incorrect identifiers, FR(correct) and FR(distracters), shows no significant difference globally or in any phrase category. This suggests that the number of fixations needed between the two identifier styles is not very different for both correct identifiers and the distracters.

With respect to the average fixation duration, AFD(Q), AFD(correct), and AFD(distracters), there is a significant difference between identifier styles over the entire data set (*p-values*=0.008, 0.015, 0.026: rows 5 through 7 in Table VI). In particular, for the question stimuli AFD(Q), 3-word code identifiers are statistically significant (*p-value* = 0.041). The distribution shows camel-cased identifiers require a higher average duration of fixations. Fig. 4 shows a gaze plot for two 3-word code identifiers showing a larger number and increased duration of fixations for the camel-case style. A fixation is shown as a circle with the radius as duration.

The average fixation duration of correct identifiers AFD(correct) is significant at the 2-word phrase (*p-value* = 0.04) and in particular for non-code identifiers (*p-value* = 0.016). See Fig. 5 for the distribution. There was no statistical significance with respect to AFD(distracters) within any identifier grouping, except for the global measure that considers all camel-cased and underscored identifiers together. Overall, based on the distribution, this suggests that for camel-cased identifiers; time taken to read the distracters is more than the underscored identifiers.

Fig. 6 shows part of a gaze plot depicting a distracter at the beginning of a 3-word non-code identifier (*mouseTheaterTicket*). Three large fixations are seen at the beginning or middle of each part (*mouse*, *Theater*, and *Ticket*) of the compound word. This indicates a longer mental parsing time needed to process the joined word.

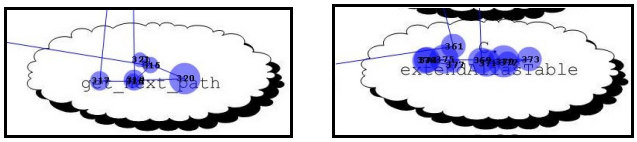

Figure 4. Part of two gaze plots for the correct underscore (left) and camel-cased (right) versions of the 3-word code identifier

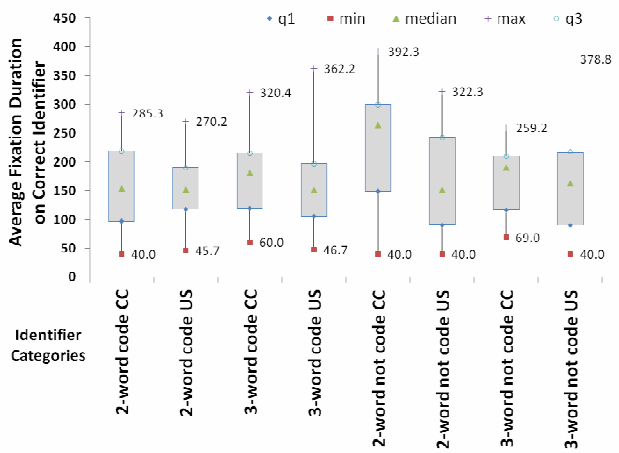

Figure 5. Descriptive statistics for AFD(correct)

TABLE VI. TWO-TAILED WILCOXON P-*VALUES* (α=0.05) FOR EACH VISUAL EFFORT MEASURE. CC=CAMEL-CASE, US=UNDERSCORE

	Grouped by Style	**Grouped by Style and Phrase Length**		**Grouped by Style, Phrase Length and Origin**			
Dependent Variable	**CC vs. US (Cohen's d)**	**2-word ident. (code ∪ non-code)**	**3-word ident. (code ∪ non-code)**	**2-word code identifiers**	**3-word code identifiers**	**2-word non-code identifiers**	**3-word non-code identifiers**
FT(Q)	0.01 * (0.57)	0.437	0.007 *	0.772	0.009 *	0.366	0.05 (~*)
FC(Q)	0.213 (0.24)	0.029 *	0.033 *	0.380	0.057 (~*)	0.004 *	0.124
FR(correct)	0.599 (0.15)	0.277	0.720	0.561	0.720	0.699	0.561
FR(distracters)	0.599 (0.15)	0.277	0.720	0.561	0.720	0.699	0.561
AFD(Q)	0.008* (0.21)	0.151	0.004 *	0.890	0.041 *	0.058 (~*)	0.277
AFD(correct)	0.015* (0.33)	0.04 *	0.208	0.679	0.277	0.016 *	0.543
AFD(distracters)	0.026* (0.21)	0.064	0.489	0.639	0.169	0.075	0.524

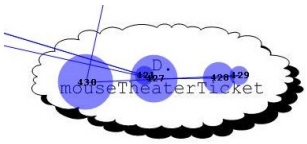

Figure 6. Part of a gaze plot for a novice showing three large fixation durations on each of the three parts of the distracter *mouseTheaterTicket*

The AFD(Q) mimics the FT(Q) measure in terms of significance in 3-word identifiers, whereas both AFD(correct) and AFD(distracters) also show significance for the 2-word identifiers but not in the 3-word category. Based on the above measures, overall, we can reject $H5_0$, indicating that visual effort is affected by identifier style. Although fixation count/rate is not significant on it's own, the average fixation duration uses the fixation count along with the time of each fixation, together having a significant effect on time required to comprehend identifiers.

Finally, after testing for interactions using two-way ANOVA, we cannot reject the null hypothesis $H6_0$ since Style does not significantly interact with Experience to have an affect on the visual effort measures.

C. *Object Memory Task*

After the main task of finding the eight identifiers was complete, subjects did a simple object memory task to detect any differences in short term memory between the two identifier styles used. This involved studying a short C++ code snippet, comprised of two methods, each 12-15 lines long, for as long as they needed. Next, a set of nine identifiers were presented and they were asked to choose which identifiers exactly matched the ones in the code snippet. There were four correct choices (two camel-case and two underscore), with the rest being distracters that changed the style or letters in the identifier. None of the subjects gave completely correct answers. On average, one of each camel-case and underscore identifier was recalled correctly but there were several false positives. This small exercise suggests no difference in recall between the two types of identifier style. This task was also conducted using the eye tracker, however eye gaze analysis of the code is beyond the scope of this paper.

D. *Similarities and Differences*

The goal of this study was the same as the Binkley et al. study [4]. Both studies use the same set of phrases to test for differences in identifier styles. The main difference between this and the previous study is the method of data collection. An eye tracker is used in this study. Conditions were also more strictly controlled with no additional personal delay biases, since the moderator advanced the screen as soon as the subject verbally stated the answer. This study was conducted as a within-subjects design where all subjects are exposed to all treatments of the factor and pair-wise comparisons are made between each identifier style. With respect to the question stimuli, the clouds are not animated as in the previous study and the phrase to be studied is not included on the question stimuli.

Another difference is the historical training received by the subjects. In this study, subjects were trained primarily in the underscore style and they were all programmers unlike the original study. An Experience variable replaced the Training variable from the original study, due to an all programmer subject sample. With respect to analyzing results, in addition to the linear mixed-model analysis (common to both studies), data is also analyzed using the non-parametric Wilcoxon test within each category due to non-normality of certain identifier groups and low sample size. In addition, an object location task is added as a post-task in this study to determine the recall ability of subjects with respect to camel-cased and underscore identifiers.

V. DISCUSSION

In response to the research questions posed in Section II, we find that identifier style significantly affects time and visual effort needed to correctly detect identifiers constructed from a phrase. The underscore style is significantly faster and positively influences the dependent variables. In the Binkley et al. study [4], Phrase Length did not interact with Style, however we find such an interaction in our analysis. The common theme in both experiments is that camel-cased identifiers take longer than underscored ones (13.5% in the previous study and 20% in this study) overall. In the Binkley et al. study, a higher accuracy was found for camel-cased identifiers, however, in our study, all (except one) subjects answered correctly on all questions making accuracy comparisons irrelevant.

In our study, no interaction effects were found between the Experience secondary factor and the independent variable Style. The previous study found Training (vis-à-vis Experience in our study) to significantly interact with Style affecting the time to find an identifier. Their findings indicate that subjects trained in the camel-case style take less time to identify a camel-cased identifier than an underscore identifier. In this study, we observed that the difference in Experience among subjects seems to interact with Style and have an effect (albeit not significant) on Find Time in two ways. First, the camel-case style shows a lesser gap between expert and novice performance (Fig. 3) and second, novices seem to benefit approximately twice as much from the underscore style than experts. Comparing this result to the original study, we can say that with more experience (training), the effect of identifier style on performance is reduced, but not eliminated.

In the Binkley et al. study, only one-third of the subjects were trained in camel-cased identifiers. In our case, we have an equal proportion of experts (8) and novices (7). During the demographic briefing, six subjects (40%) stated that they preferred camel-case, seven (47%) stated that they preferred underscore, and two (13%) had no preference. Also, in the Binkley et al. study, non-programmers stated that camel casing would be harder to visually process and thus lead to more errors. Our results prove this claim to be true with the data generated from the eye tracker (AFD(Q), AFD(correct), AFD(distracters)).

We did not look at eye fixation order, sequence of moving from one cloud to another, and the number of

regressions involved due to the relatively simple nature of the task. These measures would be more pronounced while reading a block of code versus just identifiers. The results of this study might not necessarily apply to identifiers embedded in source code. It is entirely possible that camel-cased identifiers might act as a better gestalt element when embedded inside programming constructs.

VI. Threats to Validity

Internal validity refers to the presence of other factors besides the main factor that might have an effect on the results. Since this was a within-subjects experiment we had to make sure that there was no learning effect involved when comparing the results of the two treatments for a particular phrase type. We address this by using a similar but different phrase for each identifier style. This in itself is another threat to validity, since a different phrase was used and a pair-wise comparison is made between two different phrases across the two identifier styles. Since we did not use two groups due to low sample size, we needed to make this decision. Without it, we would have seven people in each group and not much statistical power. Question order was set randomly and then fixed for each subject. Another threat to validity is the type of reading behavior subjects engaged in. Reading from top to bottom versus left to right might have an impact on how quickly subjects find an identifier. After analyzing the gaze plots for each subject, we find that all clouds were looked at to arrive at an answer.

External validity deals with generalizing our results to a real-life setting. We used students as subjects in our study, however the novices are comparable to junior software developers and experts are comparable to senior level developers since all of them have extensive programming experience. The number of our subjects appears to be low, however eye-tracking studies usually have about the same number of subjects [13]. The nature of the task used is not typical of reading code however; the basic reading process is still the same. Since this study uses the same phrases as the original study, we refer the reader there [4] for possible selection bias in terms of distracters and phrases used.

Construct validity refers to the validity of the measures used to measure performance. Since visual attention is related to mental processing of the information [17], the measures derived from the fixation counts and durations should be valid. Also, six measures for visual effort were used to avoid mono-method bias.

To ensure conclusion validity, we use the non-parametric paired Wilcoxon statistical test to determine significance due to non-normality of certain dependent variables in certain identifier groups and also less importantly due to low sample size. It is important to note that we did use ANOVA to test for interaction effects ($H6_0$), even though a few of our visual effort measures were not normally distributed. According to Wohlin et al. [26], this is possible due to the robustness of the test.

VII. Related Work

This section presents existing work on identifier names, source code readability and quality, psychology research on reading and only relevant eye tracking research related to source code and diagrams.

Lawrie et al. [18] conduct a large study on identifier names and show that actual words rather than abbreviations lead to better comprehension. Butler et al. [6] study the effect of identifier names on the quality of code. They find that identifiers that violate certain guidelines have lower code quality (more bug patterns) than ones that don't. Caprile et al. [7] study the restructuring of identifier names and the arrangement of individual words in identifiers. Binkley et al. [5] study the effect of identifier length on the recall ability of programmers, showing that longer names reduce correctness and take longer to recall. Our results in this paper add to this finding, since phrase length significantly interacts with identifier style to have an effect on performance. None of the above work considers the effect identifier style has on comprehension with the exception of [4]. The research presented here nicely complements these approaches for better identifier names.

In psychology research, Epelboim et al. [12] conducted a study on the effect fillers have on reading time. Spaces between words are filled with different fillers: Latin and Greek letters, digits and shaded boxes. They found that the type of filler had a significant effect of slowing reading speed anywhere between 10-75% depending on the filler. Shaded boxes between words (similar to underscores) had the smallest effect on reading time. Rayner et al. [21] also show decrease in reading rate by approximately 50% when fillers like *x* were used between words. Our results in this study support the above findings since a significant improvement in Find Time for underscores is shown.

Crosby and Stelovsky [8] study eye gaze data of novices and experts to determine if experience has an effect on viewing patterns. Uwano et al. [25] study eye viewing patterns of five subjects while they detect defects in source code. Their recent work focuses on multi-document review [24]. Bednarik et al. [1] study the comprehension of Java programs using eye tracking data on 18 subjects and call for more studies due to important behavior that can be revealed using eye-tracking data. They expand their study on eye tracking pair programmers simultaneously in [22]. Bednarik et al. [2] also investigate debugging behavior of 14 subjects while they debug a program in an IDE setting.

A handful of eye-tracking studies done on UML class diagrams are presented next. Yusuf et al. [27] conducted a study to determine if different class diagram layouts with stereotype information help in solving design tasks. Another study by Guéhéneuc et al. [15] uses an eye tracker to investigate how designers answer two simple questions about modifying parts of the diagram. They also study the effect of the presence of the Visitor pattern in class diagrams [16].

VIII. Conclusions and Future Work

An eye-tracking study analyzing the effect of identifier style (camel-case and underscore) on accuracy, time, and visual effort is presented with respect to the task of recognizing a correct identifier, given a phrase. Visual effort is determined using six measures based on eye gaze data namely: fixation counts and durations. Although, no

difference was found between identifier styles with respect to accuracy, results indicate a significant improvement in time and lower visual effort with the underscore style. The interaction of Experience with Style indicates that novices benefit twice as much with respect to time, with the underscore style. This implies that with experience or training, the performance difference between styles is reduced. These results add to the findings of Binkley et al.'s study [4]. Future work includes conducting more eye-tracking studies (with a larger subset of identifiers and larger subject sample), on reading source code consisting of both identifier styles, in the context of a specific task such as debugging. Another possible direction is to determine if there is an advantage for a programmer to change their current style to what is determined to be a better overall style.

ACKNOWLEDGMENT

We would like to thank Dr. David Robbins for assisting in the use of the Tobii eye tracker and all the people who took the time to participate in this study.

REFERENCES

[1] Bednarik, R. and Tukiainen, M., "An Eye-tracking Methodology for Characterizing Program Comprehension Processes", in Proceedings of Symposium on Eye tracking research &Applications(ETRA),California,2006,pp.125-132.

[2] Bednarik, R. and Tukiainen, M., "Temporal Eye-tracking Data: Evolution of Debugging Strategies with Multiple Representations", in Proceedings of Symposium on Eye Tracking Research & Applications (ETRA), Savannah, Georgia, 2008, pp. 99-102.

[3] Beymer, D. and Russell, D. M., "WebGazeAnalyzer: a system for capturing and analyzing web reading behavior using eye gaze", in Proceedings of CHI '05 extended abstracts on Human factors in computing systems, Portland, OR, USA, 2005, pp. 1913-1916.

[4] Binkley, D., Davis, M., Lawrie, D., and Morrell, C., "To CamelCase or Under_score", in Proceedings of 17th IEEE International Conference on Program Comprehension (ICPC), Vancouver, Canada, 2009, pp. 158-167.

[5] Binkley, D., Lawrie, D., Maex, S., and Morrell, C., "Identifier Length and Limited Programmer Memory", Science of Computer Programming, vol. 74, 2009, pp. 430-445.

[6] Butler, S., Wermelinger, M., and Sharp, H., "Relating Identifier Naming Flaws and Code Quality: An Empirical Study", in Proceedings of 16th Working Conference on Reverse Engineering (WCRE), Lille, France, Oct 2009, pp. 13-16.

[7] Caprile, B. and Tonella, P., "Restructuring Program Identifier Names", 16th IEEE International Conference on Software Maintenance (ICSM) 2000, pp. 97-107.

[8] Crosby, M. E. and Stelovsky, J., "How Do We Read Algorithms?A Case Study", IEEE Computer, v.23, n.1, 1990, pp. 24-35.

[9] Cutrell, E. and Guan, Z., "What Are You Looking For? An Eye-tracking Study of Information Usage in Web Search", in Proceedings of CHI, San Jose, CA, USA, 2007, pp. 407-416.

[10] de Kock, E., van Biljon, J., and Pretorius, M., "Usability Evaluation Methods: Mind the Gaps", in Proceedings of Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists Vanderbijlpark, Emfuleni, South Africa 2009, pp. 122-131.

[11] Deißenböck, F. and Pizka, M., "Concise and Consistent Naming", Software Quality Journal, vol. 14, no. 3, 2006, pp. 261-282.

[12] Epelboim, J., Booth, J. R., Ashkenazy, R., Arash, T., and Steinman, R. M., "Fillers and Spaces in Text: The Importance of Word Recognition During Reading", Vision Research, vol. 37, no. 20, 1997, pp. 2899-2914.

[13] Goldberg, J. H. and Kotval, X. P., "Computer Interface Evaluation Using Eye Movements: Methods and Constructs", International Journal of Industrial Ergonomics, vol. 24, 1999, pp. 631-645.

[14] Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., and Wichansky, A. M., "Eye tracking in web search tasks: design implications", in Proceedings of 2002 symposium on Eye tracking research & applications (ETRA), New Orleans, Louisiana, 2002, pp. 51-58.

[15] Guéhéneuc, Y.-G., "TAUPE: towards understanding program comprehension", in Proc. of 16th IBM Centers for Advanced Studies on Collaborative research, Canada,Oct 2006, pp.1-13

[16] Jeanmart, S., Guéhéneuc, Y.-G., Sahraoui, H., and Habra, N., "Impact of the Visitor Pattern on Program Comprehension and Maintenance", in Proc. of 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, Florida, Oct 15-16 2009, pp. 69-78.

[17] Just, M. and Carpenter, P., "A Theory of Reading: From Eye Fixations to Comprehension", Psychological Review, vol. 87, 1980, pp. 329-354.

[18] Lawrie, D., Morrell, C., Feild, H., and Binkley, D., "What's in a Name? A Study of Identifiers",14th Intl. Conference on Program Comprehension (ICPC) 2006, pp.3-12.

[19] Matsuda, Y., Uwano, H., Ohira, M., and Matsumoto, K.-i., "An Analysis of Eye Movements during Browsing Multiple Search Results Pages ", in Lecture Notes in Computer Science, Springer Berlin, 2009, pp. 121-130.

[20] Nakamichi, N., Shima, K., Sakai, M., and Matsumoto, K.-i., "Detecting Low Usability Web Pages using Quantitative Data of Users' Behavior", in Proceedings of 28th International Conference on Software Engineering (ICSE'06), Shanghai, China, 2006, pp. 569-576.

[21] Rayner, K., Fischer, M. H., and Pollatsek, A., "Unspaced Text Interferes with Both Word Identification and Eye Movement Control", Vision Research, vol. 38, no.8,1998,pp.1129-1144.

[22] Sami, P., Roman, B., Tatiana, G., Vesa, T., and Markku, T., "A Method to Study Visual Attention Aspects of Collaboration: Eye-tracking Pair Programmers Simultaneously", in Proceedings of Symposium on Eye Tracking Research & Applications, Georgia,2008, pp. 39-42.

[23] Sutter, H. and Alexandrescu, A., C++ Coding Standards:101 Rules,Guidelines, and Best Practices, Addison-Wesley,2004.

[24] Uwano, H., Monden, A., and Matsumoto, K.-i., "DRESREM 2: An Analysis System for Multi-document Software Review Using Reviewers' Eye Movements", in Proceedings of 3rd International Conference on Software Engineering Advances (ICSEA), Sliema, Malta, 2008, pp. 177 - 183

[25] Uwano, H., Nakamura, M., Monden, A., and Matsumoto, K., "Analyzing Individual Performance of Source Code Review Using Reviewers' Eye Movement", in Proceedings of 2006 symposium on Eye tracking research & applications (ETRA), San Diego, California, 2006, pp. 133-140.

[26] Wohlin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., and Wesslen, A., Experimentation in Software Engineering - An Introduction., Kluwer Academic Publishers, 2000.

[27] Yusuf, S., Kagdi, H., and Maletic, J. I., "Assessing the Comprehension of UML Class Diagrams via Eye Tracking", in Proc. of 15th IEEE Intl. Conf. on Program Comprehension (ICPC), Banff Canada, June 26-29 2007, pp. 113-122.