Ilkogretim Online - Elementary Education Online, 2021; 20 (3): pp. 1035-1041
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.03.113

A Study on Effects of Code Review on Reasoning of Aviation Engineers

***Jae-Yong Lee,** *Professor, Dept. of Unmanned Aircraft Systems, Hanseo Univ.(TAean Campus), Gomseom-ro, Nam-Myeon, Taean-Gun, Chungcheongnam-do, 32158, Republic Of Korea, , jylee@hanseo.ac.kr*
*Corresponding author

Abstract. We aim to identify how code review, which helps integrate information in C programming learning, affects the knowledge of reasoning divided into three categories. Thirty-eight college students studying aeronautical engineering were classified as "understanding and experience in reasoning overview", "understanding and experience in reasoning classification", and "expectation in use of reasoning", and the effects of code review were then analyzed. Reliability analysis, descriptive statistical analysis, correlation analysis, and regression analysis were performed using Pandas. Based on the results of Pearson's Correlation Coefficient between the questions on the knowledge of reasoning and Pearson's Correlation Coefficient between questions on code review, the correlation level of code review was lower than that of knowledge of reasoning. The data learning results showed an a-value of 16.455 and a b-value of 0.579 in the first survey, an a-value of 16.195 and a b-value of 0.700 in the second survey, and an a-value of 11.729 and a b-value of 0.829 in the third survey. The performance evaluation results showed that the MSEs were 81.171, 58.854, and 54.029 in the first, second, and third surveys, respectively. In addition, the RMSEs were 9.009 for the first survey, 7.671 for the second survey, and 7.350 for the third survey. For the hypothesis testing of this study, the significance probability of .024 was found in the first survey, .009 was found in the second survey, and .041 was found in the third survey, indicating that it was significant in all three surveys

Keywords: Programming Interest, Code Review, Bottom-up Processing, Top-down Processing, Psychology of Programming

Received: 04.12.2020	Accepted: 10.01.2021	Published: 06.02.2021

INTRODUCTION

The programming process is a mental process of thinking simulation[1]. The program, which is the outcome of the mental process of a person, is the result of cooperation-based work among a few programmers[1]. As a result, it is difficult to measure the knowledge activities that take place in the process of the program, and therefore, related research has not been developed for a long time[2]. To evaluate the effectiveness or outcome of programming production, studies should investigate the relevance of completeness between people and people or people and programming[3]. Efforts to interpret programming as a result of collaborative work among programmers, a method of cooperation among a few programmers, and cognitive processes of programming as psychological problems are developing in the psychology of programming[4,5].

At the same time, as the society of the fourth industrial revolution based on intelligence information is approaching, society is rapidly shifting away from providing programming education only to computer engineers to ensuring that talented people in all academic fields can program A.I. There are many difficulties associated with applying the curriculum of traditional computer technicians to other fields[6,7]. For example, all students of the Department of Aeronautics are also required to take basic SW·AI education to prepare for the intellectualized and informationized society. Aeronautics is an academic field that requires training professionals in piloting, manufacturing, and maintenance, with a focus on aerodynamics. Aerospace engineers treat mathematical knowledge of differentials and integrals as a basic subject, and they understand and learn problems based on the differentials. Artificial intelligence engineering programming, which involves directly programming differential and integral calculus, can be very advantageous for aerospace engineers to learn artificial intelligence programming. These features will enable the information processing abilities to be equipped without requiring an extensive computer science curriculum. In other words, by teaching individuals to rapidly integrate the skills required for deductive and inductive thinking simulation and ensuring the completion of AI programming courses based on concepts of matrix as well as differential and integral calculus and statistics, it would be possible to achieve aviation-based intelligent information processing abilities through a short curriculum.

By shortening the period of knowledge training and finding a programming education model that allows direct programming of matrix as well as differential and integral calculus and statistics, the deductive thinking training required throughout SW education and inductive thinking training of AI education will be able to be effectively taught. To this end, the formation of a mental model of programming psychology, the use of metaphors, and code review are known to be effective methods. Among these, code review, particularly from the viewpoint of programming psychology, is known to be very useful in integrating knowledge information and enhancing the reasoning ability in the process of reading, writing, learning, and solving problems[8]. For the integration of programming creating knowledge, there needs to be a study examining whether the level of knowledge simulation affects the knowledge related to the reasoning. Specifically, it is necessary to find a methodology to help thinking simulation by engineering groups, such as aviation engineers, who have characteristics that are difficult to assign all subjects to the training of thinking simulation, or that are difficult to spend long hours on. For example, it is not known how much code review increases the knowledge of reasoning, deductive thinking, and inductive thinking, which is the ultimate goal of computational thinking and artificial intelligence programming.

1. Theory and formula

The nature of the programming process abstracts the problem and embodies it as a verbal action called programming under high cognitive loads. The programming process is conducted using bottom-up processing and top-down processing of senses and perception from the inner side of humankind. Information integration of these two processes takes place during computer programming. It is known that the code review leads to both bottom-up and top-down processing[9,10].

2. Method of Study

This thesis identifies how information integration through code review affects the knowledge of reasoning. The survey for code review consisted of "Overview of Code Review" (CR1), "Language Use in Code Review" (CR12), and "Language Comprehension in Code Review" (CR3). The survey on the degree of awareness of reasoning consisted of "Comprehension of the Reasoning Overview" (IF1), "Comprehension and Experience of Reasoning Classification" (IF2), and "Expectation for the Use of Reasoning" (IF3). Table 1 presents the survey used for this object.

Table 1 : Programming Inference and Code review Questionnaire

Inference questionnaire	Comprehension of the Reasoning Overview (IF1)	Concept of Reasoning	IQ1: I know that the programming process is related to reasoning.
		Reasoning Classification	IQ2: I know that programming is a problem with deductive, inductive, and analogical reasoning.
	Comprehension and Experience of Reasoning Classification (IF2)	Deductive Reasoning	IQ3: I have used deductive reasoning in the C programming process.
		Inductive Reasoning	IQ4: I have used inductive reasoning in the C programming process.
		Analogical Reasoning	IQ5: I have used analogical reasoning in the C programming process.
	Expectation for the Use of Reasoning (IF3)	Deductive Method	IQ6: I hope there are many opportunities to use deductive methods when programming.
		Inductive Method	IQ7: I hope there are many opportunities to use inductive methods when programming.
		Analogical Reasoning	IQ8: I hope there are many opportunities to use analogical reasoning when programming.
Code review questionnaire	Overview of Code Review (CR1)	Interest in Code Review	CQ1: It is new and interesting to do a code review for programs created by friends in each class.
		Information	CQ2: I have experienced 'Aha!' when I was doing a code review

		Integration	during the class.
	Language Use in Code Review (CR2)	Code Reading	CQ3: After code review, I could do programming reading better during the class.
		Code Writing	CQ4: After code review, I could do programming writing better during the class.
	Language Comprehension in Code Review (CR3)	Code Listening	CQ5: After code review, I could do programming listening better during the class.
		Code Grammar	CQ6: After code review, I could do programming grammar better during the class.

A survey was conducted among 38 college students in the Department of Aeronautics who are beginners in learning C programming (N=38). The response distribution of the questionnaire and the general characteristics of the survey subjects were identified through a descriptive statistical analysis. In addition, a reliability analysis was conducted to confirm the reliability of the question. A correlation analysis was also performed to identify the correlation between code review and reasoning. Finally, a regression analysis was conducted to examine the relationship between code review and reasoning. These experiments were conducted using pandas.

3. Result and Discussions

A descriptive statistical analysis was conducted to measure the code review levels of aerospace engineers. The questionnaire measuring the code review level consists of six questions answered on a seven-point scale. In the code review level of the first survey, the deviation was -0.571, the kurtosis was 0.383, the average was 30.632, and the standard deviation was 6.859. In the second survey, the deviation was -0.849, the kurtosis was 0.552, the average was 31.184, and the standard deviation was 7.097. In the third survey, the deviation was -0.812, the kurtosis was 0.634, the average was 32.263, and the standard deviation was 7.395.

A descriptive statistical analysis was conducted to measure the reasoning levels of aerospace engineers. The questionnaire measuring the reasoning level consists of eight questions. The reasoning levels were measured three times after code review. In the reasoning level of the first survey, the deviation was -0.275, the kurtosis was -0.052, the average was 34.184, and the standard deviation was 9.956. In the reasoning level of the second survey, the deviation was -0.017, the kurtosis was -0.740, the average was 37.079, and the standard deviation was 9.113. In the reasoning level of the third survey, the deviation was -0.297, the kurtosis was -0.676, the average was 38.474, and the standard deviation was 9.647.

In the first survey of the reasoning surveys, the Cronbach alpha in the "Comprehension of the Reasoning Overview" (IF1) was 0.893, the Cronbach alpha in the "Comprehension and Experience of Reasoning Classification" (IF2) was 0.978, and the Cronbach alpha in "Expectation for the Use of Reasoning" (IF3) was 0.907. In the second survey of the surveys for reasoning, the Cronbach alpha in the "Comprehension of the Reasoning Overview" (IF1) was 0.778, the Cronbach alpha in the "Comprehension and Experience of Reasoning Classification" (IF2) was 0.971, and the Cronbach alpha in "Expectation for the Use of Reasoning" (IF3) was 0.889. In the third survey of the surveys for reasoning, the Cronbach alpha in "Comprehension of the Reasoning Overview" (IF1) was 0.873, the Cronbach alpha in the "Comprehension and Experience of Reasoning Classification" (IF2) was 0.953, and the Cronbach alpha in "Expectation for the Use of Reasoning" (IF3) was 0.989.

In the first survey of the surveys for code review, the Cronbach alpha in "Overview of Code Review" (CR1) was 0.774, the Cronbach alpha in "Language Use in Code Review" (CR2) was 0.924, and the Cronbach alpha in "Language Comprehension in Code Review" (CR3) was 0.928. In the second survey of the surveys for code review, the Cronbach alpha in "Overview of Code Review" (CR1) was 0.705, the Cronbach alpha in "Language Use in Code Review" (CR2) was 0.971, and the Cronbach alpha in "Language Comprehension in Code Review" (CR3) was 0.944. In the third survey of the surveys for code review, the Cronbach alpha in the "Overview of Code Review" (CR1) was 0.815, the Cronbach alpha in "Language Use in Code Review" (CR2) was 0.978, and the Cronbach alpha in "Language Comprehension in Code Review" (CR3) was 0.956.

Therefore, the reliability of the question could be verified according to the highest order of Cronbach alpha value. However, due to the lack of an exact operational definition in the code review questions for the reasoning level, the problems among questions have not been considered by focusing on the surveys described in various theses. In addition, code review also showed information integration, reading, writing, listening, grammar, and high reliability, but there was also a lack of a proper operational definition that integrates bottom-up and top-down processing.

The results of the Pearson correlation analysis of the first survey among the questions for reasoning are presented in <Table 2>, and those for the Pearson correlation analysis of the code review are listed in <Table 3>. As shown in <Table 2> and <Table 3>, the correlation of code review was lower than that of reasoning. Regarding the P-values among the questions of the reasoning, < .0.05 was only found for one value while the rest were < .01. This was also the case in the correlation of the code review, as only one p-value was <.05 while the rest were <.01. Similar results were shown in the second and third surveys.

Table 2: Correlation of Surveys on Reasoning (First survey)
(N:38)

	IF10	IF11	IF12	IF13	IF14	IF15	IF16	IF17
IF10	1							
IF11	0.807494**	1						
IF12	0.783322**	0.616113**	1					
IF13	0.774048**	0.603285**	0.936316**	1				
IF14	0.725069**	0.556697**	0.910474**	0.962408**	1			
IF15	0.697486**	0.563216**	0.63327**	0.689545**	0.670324**	1		
IF16	0.564888**	0.336685*	0.47654**	0.501605**	0.470925**	0.659074**	1	
IF17	0.712785**	0.567585**	0.690815**	0.716847**	0.713593**	0.940589**	0.694367**	1

*** : p < .05 ** : p < .01**

Table 3: Correlation of Surveys on Code Review (First survey)
(N:38)

	CR10	CR11	CR12	CR13	CR14	CR15
CR10	1					
CR11	0.640074**	1				
CR12	0.538102**	0.588427**	1			
CR13	0.472096**	0.432137**	0.863423**	1		
CR14	0.420566**	0.355564*	0.821195**	0.896355**	1	
CR15	0.605083**	0.577704**	0.900746**	0.872007**	0.866459**	1

*** : p < .05 ** : p < .01**

The results of the study showed that the a-value of the first survey was 16.455 and the b-value was 0.579. The a-value of the second survey was 16.190 and the b-value was 0.700. The third survey showed an a-value of 11.729 and a b-value of 8.829, as presented in Table 4. The regression analysis result for the first survey is shown in <Figure 1>, that for the second survey is shown in <Figure 2>, and that for the third survey is shown in <Figure 3>.

Table 4: Analysis Results

(N:38)

	First survey	Second survey	Third survey
P values	0.024	0.009	0.041

R_squared	0.159	0.272	0.404
a value	16.455	16.190	11.729
b value	0.579	0.700	0.829
Mean Squared Error	81.171	58.854	54.029
RMSE	9.009	7.671	7.350

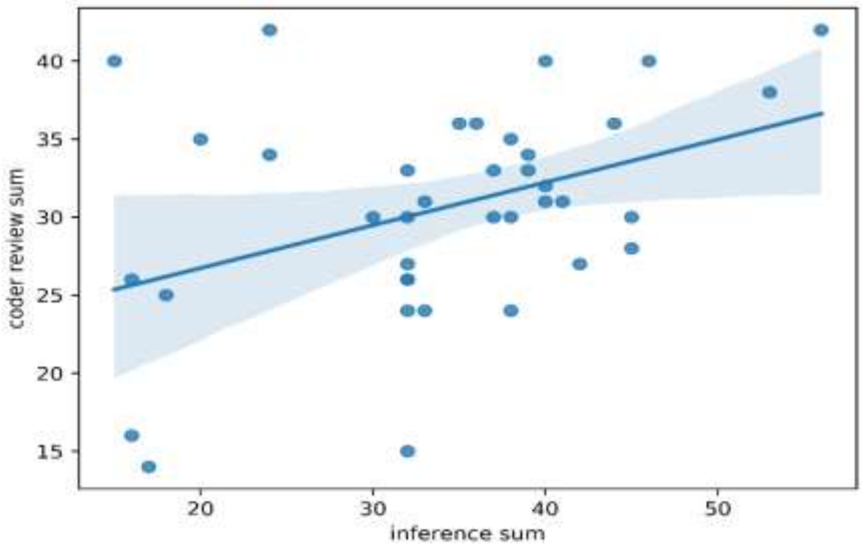

Figure 1: Scatter plot of total inference and code review after first code review

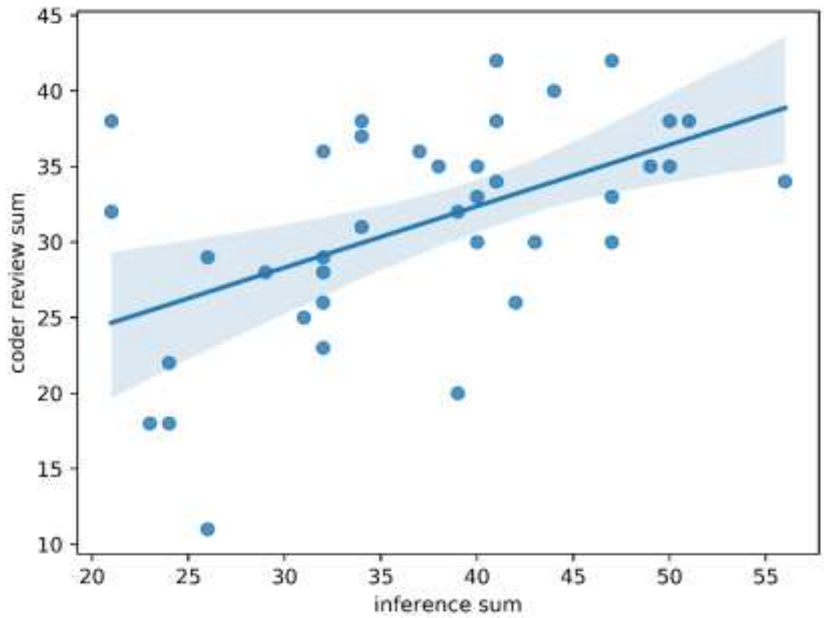

Figure 2: Scatter plot of total inference and code review after second code review

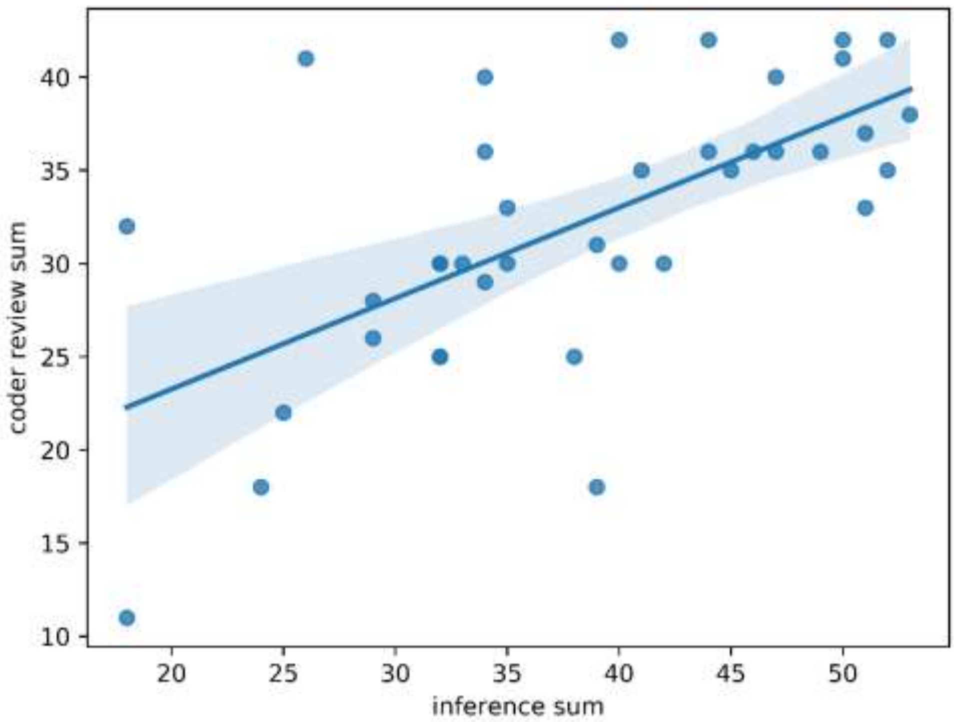

Figure 3: Scatter plot of total inference and code review after third code review

Regarding the performance evaluation results, the Mean Squared Errors were 81.172 in the first survey, 58.854 in the second survey, and 54.030 in the third survey. In addition, the RMSEs were 9.010 in the first survey, 7.672 in the second survey, and 7.350 in the third survey.

For the hypothesis testing of this study, the significance probabilities were .024 in the first survey, .009 in the second survey, and .042 in the third survey. This indicates that significant results were obtained in all three surveys.

4. Conclusions

Aerospace engineers directly apply the concept of differential and integral calculus in their work, which is advantageous for creating high-level programs. Nevertheless, C programming takes a long time to understand pointer, object, and high-level abstract concepts and to integrate complex cognitive concepts, thus making it difficult to study. In this thesis, aerospace engineers measured the extent to which the comprehension of reasoning increased by repeating the code review. To this end, the impact of the code review on the reasoning of 38 college students in the Department of Aeronautics was analyzed. For this purpose, the knowledge of reasoning was classified as "Comprehension of the Reasoning Overview" (IF1), "Comprehension and Experience of Reasoning Classification" (IF2), or "Expectation for the Use of Reasoning" (IF3) to verify that the constituting operational definition is applicable for the knowledge of reasoning. For the code review, the survey was organized by dividing it into "Overview of Code Review" (CR1), "Language Use in Code Review" (CR2), and "Language Comprehension in Code Review" (CR3).

Through this analysis, the correlation with code review and knowledge of reasoning was identified, as was the impact through the regression analysis. The measurements were taken three times during the class while conducting a code review.

All the questions that classified knowledge of reasoning had reliable levels, but further study is needed for inter-item analysis. In the measurement of the knowledge of reasoning and the correlation with code review, the correlation of all three measurements was found to be significant. The effect of the code review on the knowledge of reasoning could be seen to have a greater effect over time. The results confirmed that the code review is an effective way for aerospace engineers to improve their reasoning ability, which is the result of the integration of cognitive abilities. In addition, it can be seen that a systematic study on the effects of code review on other programming capabilities is necessary.

Acknowledgements

This study was conducted with the support of the 2020 Hanseo University Intramural Research Support Project

References

[1] F. B. Alan, (2002) What is Programming?. *14th Workshop of the Psychology of Programming Interest Group*, London, UK, 204-218.
[2] R.D. Pea, D. M. Kurland D.M., (1984) "On the cognitive effects of learning computer programming," *New Ideas in Psychology* 2(2), 137-168.
[3] Martin H., (1991) *Handbook of Human-Computer Interaction.* 2nd ed. Elsevier science publishing company inc., Amsterdam, chapter 1.
[4] Hoc J.M., Green T.R.G., Samuçay R., and Gilmore D.J., (2017) *Psychology of Programming.* London, ACADEMIC PRESS, chapter 1.
[5] Sultana, S. (2016) *Defining the Competencies, Programming Languages, and Assessments for an Introductory Computer Science Course.* Ph. D. Desertation, Old Dominion University, U.S.A.
[6] de Raadt M., (2007) "A Review of Australasian Investigations into Problem-Solving and the Novice Programmer," *Journal of Computer Science Education* 17(3), 201-213.
[7] R. E. Mayer, (1975) "Different problem-solving competencies established in learning computer programming with and without meaningful models," *Journal of Educational Psychology* 67(6), 725-734.
[8] D. Teague and R. Lister, (2014) Programming: reading, writing and reversing, Proc. of the 2014 conference on Innovation & technology in computer science education, Uppsala Sweden, 285-290.
[9] B. Mario, M. Andreas and G. Thomas, (2010) Adopting Code Reviews for Agile Software Development, Proc. of 2010 Agile Conference, Orlando U.S.A., 9-13.
[10] J. Y. Lee, (2019) "A Study on the Correlation between Code Review and Computer Programming Interest," *Journal of Advanced Research in Dynamical and Control Systems (JARDCS)* 11(05), 2434-2438.