

The Effect of Flipped Learning on Problem-Solving Capability in Software Education

Ji Hyun Park and Tae In Han

Abstract—The present study attempted to examine the effect of software education on the learning outcome of non-major learners when flipped learning is applied to the software education of non-major college students. The study set problem-solving as the sub-variables of learning outcome. To do this, 50 people, two classes of software course students, participated in the experiment. Experimental group applied the flip learning to the eight classes for four weeks and comparative group applied the general class of the traditional classroom.

The conclusions obtained from the research results and discussions are as follows.

Software learning for non-majors, which applied flipped learning, is more effective for improving learning outcomes (problem-solving capability) than general classes. Out of the problem-solving ability sub-variables, it is particularly effective in improving decision-making. This allows the interpretation that decision making, among the problem-solving ability sub-variables of learners, must have improved because software education that applied flipped learning is carried out by actively reflecting the learner's activities based on problem-solving.

Index Terms—Flipped learning, learning effect, problem solving, software education.

I. INTRODUCTION

A. Research Goals and Objectives

Software education comprises a series of thinking-based education that collects and analyzes data based through computers to find solutions for various problems and creates an effective solution process. The Korean Ministry of Education and the Ministry of Science, ICT, and Future Planning introduced a blueprint for software education from elementary school to university through the Human Resource Development Promotion Plan for Software-Oriented Society, which is designed to develop the core competencies required for success in the 21st century. In addition, it revised the elementary and secondary curriculum to provide software education as the basis for computing thinking [1], [2].

Software education can be subject to individual differences such as individual aptitude or level of interest. Even if students receive the same education, there will be some who are more creative in applying what they have learned while others will experience difficulty in understanding the basics. In addition, coding in software education is not fixed, and various methods and results are derived according to individual cognition processes. Because

of the nature of this software training, providing software education uniformly without variation to all students in a classroom setting while remaining equally effective to every student is very challenging [3].

As a result, for software education, teaching according to the level of each individual student has become necessary and, accordingly, various teaching methods are being studied in response to this imperative. Recently, flipped learning, a form of blended learning in which classroom instruction and e-learning technology are integrated, is getting attention for providing effective classroom instruction [4].

In this paper, we select problem solving ability as a learning variable in the software course, classified as a university liberal arts course, and investigate the effect of problem solving ability on flipped learning.

B. Research Problems and Hypotheses

- Research question: What factors influence learner problem-solving ability in flipped learning-based software education?
- Research hypothesis: The group with flipped learning will have more problem solving ability than a comparative (control) group with traditional general instruction in software education.

II. BACKGROUND

A. Concepts and Features of Flipped Learning

Flipped learning is a pedagogical approach in which the conventional notion of classroom-based learning is inverted. Learning materials are provided to students online before the class session and in-class time is devoted to exercises, projects, or discussions based on the learning materials. During class sessions, instructors function as coaches or advisors, and provide mixed learning centering on the interaction of learner and learner, and interaction of teacher and learner [5].

The characteristic of flipped learning is to build a learner-centered classroom. Teachers can focus on their interaction with learners and help them with difficulties in problem solving so they can engage in more intensive learning activities [6].

B. Effect of Flipped Learning

The flipped learning application for calculus education at was improving instructional method in order to make the learners better effective understanding about mathematical concept at Michigan University in USA. It performed brief lecture and discussion learning about what learners read the resources which was provided by instructor before classroom

Manuscript received August 1, 2017; revised September 12, 2017.

The authors are with the Department of e-Learning Graduate School, Korea National Open University, Seoul, Korea (correspondence author: Tae In Han; e-mail: com-teacher@nate.com).

learning [7].

The pupil of primary school had the good custom of self-directed learning behavior after flipped learning experience for science education at one of the Korean elementary school. And furthermore, it was also analyzed that flipped learning contributed to the learners to get more learning achievement and motivation than learners of traditional classroom learning by statistical result of significant level [8].

C. Software Education

Software education at universities was initiated by the Software-Centered University Project in 2015 to revolutionize college software education. At J University, this is being extended to students who are not majoring in computer science by utilizing its existing Basic Liberal Arts Education Institute [9].

In addition, as there is a steady increase in interest in and demand for software education among college students, this program allows students not majoring in computer science to broaden their opportunities in receiving software education and thus become software human resources able to integrate their respective major with software education.

D. Creative Problem Solving

Creative Problem Solving (CPS) is a cognitive process based on knowledge and function, in which divergent and convergent thinking interact to produce new products or solutions [8]. Intended to help effective problem solving when individuals are confronted with ambiguous and complex problems in everyday life, it enables students to investigate factors requisite for problem solving, conceive creative ideas and think creatively while experiencing their own problem solving processes.

In this study, CPS is divided into sub-domains, namely problem finding, idea generation, and solution evaluation. The higher the sum of the scores in each sub domain, the higher the creative problem solving ability.

III. RESEARCH METHODS

A. Research Subjects and Experimental Design

This study was conducted on students not majoring in computer science at one of the university in Jeollabuk-do, Korea. Among the students taking the liberal arts software course, two groups of 25 were formed, one group was the experiment group that performed the flipped learning, and the other group was comparative (control) group that followed the traditional teaching method.

The students of two groups were in same major and same grade (sophomore), and they had no experience in learning about software education. This study used the Scratch software as a subject of software education because it was applied frequently on software education for creative problem solving abilities.

The pre-test and post-test control group for heterogeneity was designed as shown in Table I.

B. Flipped Learning Scenario

For applying flipped learning, the software educational

program was designed and developed by two types of on-line learning and classroom learning for this study. The students of experiment group had to learn the on-line learning before classroom learning, and they performed discussion learning about what they learn from on-line learning in the classroom, but the students of control group learned the lesson by instructor’s lecture in the classroom directly. Instructional design for two class is shown Table II.

TABLE I: EXPERIMENTAL DESIGN

Group	Pre-test	Experimental Treatment	Post-test
Experiment group	O1	X1	O4
Comparative group	O2	X2	O5

O1, O2: Pre-test(Creative Problem Solving; CPS)
 X1: Flipped learning based software education
 X2: Software education in traditional general classes
 O4, O5: Post-test(Creative Problem Solving; CPS)

Two class were operated continuously at the same time in the different class during one semester. And individual survey about the creative problem solving abilities for learners were conducted before and after the software education class.

TABLE II: INSTRUCTIONAL DESIGN FOR TWO DIFFERENT LEARNING GROUP ABOUT BEFORE, IN AND AFTER CLASS

Class		Flipped learning	Traditional learning
Before Class	teacher	- Strategy - Producing video lecture - Preparing discussion learning	- Preparing class - Producing lecture
	student	- Learning video - Preparing question	- Doing assignment - Pre-study the lecture
In Class	teacher	- Question and answer - Suggest mission - Discussion activities - Summary - Mini lecture - Feedback	- Suggest learning goal - Motivation - Tutoring - Question and answer - Summary - Assignment - Introducing next lecture
	student	- Response journal for class - Reflection for class	- Preparing next lesson
After Class	teacher	- Response journal for class - Reflection for class	- Preparing next lesson
	student	- Summary note - Suggestion for class - Sharing information	- Doing assignment

IV. MEASUREMENT METHODS

A. Creative Problem Solving

Creative problem solving for the computer software education required many related abilities according to this research. The Korean government published the major abilities for creative problem solving for software education in 2015 [10], this study applied governmental suggestion because this program should be operated in normal higher education institute (university). According to those suggestion and related research, a total of 26 questions were

constructed and each was measured on a 5-point Likert scale [11].

B. Three Factors of Creative Problem Solving

This study identified three factors of creative problem solving in order to classify the different question items of characteristics in the questionnaire for creative problem solving. Therefore this study sub-categorized three factors which are related ability of information gathering, divergent thinking and planning from the 26 questions. For this classification, this study used correlation and factor analysis, and it made three factors as sub-categories for creative problem solving.

Table III shows what question classified to sub-categories (factors) from question 1 to question 26.

TABLE III: COMPOSITION OF QUESTIONS FOR SUB-CATEGORIES OF CREATIVE PROBLEM SOLVING

Sub-Categories	Questions Number	Number of Questions
Ability to analyze information gathering	1-10	10
Divergent thinking based decision making	11-20	10
Planning ability	21-26	6
Sum of the scores		26

C. Survey Data Gathering and Statistical Analysis for the Research Hypothesis

Survey data were gathered from all students of two classes before and after the software education class which was continued one semester (15 weeks).

This study used Cronbach’s α test for reliability test about three factors, and used t-test in order to verify difference between experimental group and control group for three factors which was related to creative problem solving, it involved the ability to analyze information gathering, the divergent thinking based decision making and the planning ability.

For analysis tool, this study used SPSS (Statistical Package for Social Sciences) v23.0.

A. Cronbach’s α Test for Reliability about Questionnaire of Item

First above all, scale reliability test about three factors was performed by the Cronbach’s α test in order to validate the items of variables which are consisted of three factors. The result of test is shown in Table IV. According to the result, all of the variables of factors were confidential and reliable (value of Cronbach’s α was more than 0.70) to represent the characteristic for information gathering, divergent thinking, and planning abilities.

B. Homogeneity Test for Experimental Group and Comparative Group

The two groups, experimental and control, exhibited similar result of pre-test, it means that the result was statistically insignificant, and research hypothesis could not be accepted. For this analysis, scores in the three sub-factors of creative problem solving before flipped learning was applied, as shown in Table V.

TABLE IV: CRONBACH’S A TEST RESULT ABOUT VARIABLES OF THREE FACTORS

Factors	Questions Number	Cronbach’s α value
Ability to analyze information gathering	1-10	0.74
Divergent thinking based decision making	11-20	0.82
Planning ability	21-26	0.76

C. Flipped Learning Based Software Education and Creative Problem Solving

Analysis of the differences between the pre-test and post-test results of the experimental group and comparison group revealed that flipped learning had a significant effect on creative problem solving as shown in Table VI. It seems that the method of flipped learning, in which software basics are presented in advance through the class materials provided by the teacher before class, and students understand according to their individual level through various activities and discussions in the classroom, helped students to understand the contents of the class more effectively.

V. RESULTS

TABLE V: T-TEST IN PRE-TEST FOR CPS OF EXPERIMENT GROUP AND COMPARISON GROUP

Sub-Variables	Group	N	M \pm SD	df	t
Ability to analyze information gathering	Experimental Group	25	3.33 \pm 0.50	54	0.947
	Comparative Group	25	3.20 \pm 0.46		
Divergent thinking based decision making	Experimental Group	25	3.36 \pm 0.47	54	1.047
	Comparative Group	25	3.22 \pm 0.46		
Planning ability	Experimental Group	25	3.21 \pm 0.53	54	-0.490
	Comparative Group	25	3.28 \pm 0.56		

TABLE VI: T-TEST IN CREATIVE PROBLEM SOLVING

Sub-Variables	Group	Experimental Treatment	N	M±SD	t	Pr > t
Ability to analyze information gathering	Experimental Group	Pre-test	25	3.33±0.50	3.156**	0.01
		Post-test	25	3.74±0.76		
	Comparative Group	Pre-test	25	3.20±0.46		
		Post-test	25	2.93±0.73		
Divergent thinking based decision making	Experimental Group	Pre-test	25	3.36±0.47	3.106**	0.01
		Post-test	25	3.04±0.55		
	Comparative Group	Pre-test	25	3.22±0.46		
		Post-test	25	3.16±0.56		
Planning ability	Experimental Group	Pre-test	25	3.21±0.53	4.368**	0.001
		Post-test	25	3.81±0.62		
	Comparative Group	Pre-test	25	3.28±0.56		
		Post-test	25	3.27±0.54		

*: Significance level 0.05 **: Significance level 0.01 ***: Significance level 0.001

Therefore, the research hypothesis, i.e., the experimental group that applied flipped learning in software education will have more creative problem solving ability than the comparative group that applied traditional education, was accepted.

D. Conclusion

This study started from the research question “What factors influence learner problem-solving ability in flipped learning-based software education?”, and in order to resolve this question, this study had the research hypothesis of “The group with flipped learning will have more problem solving ability than a comparative (control) group with traditional general instruction in software education.”

In conclusion, this study did not discovered and determined the factors or elements which inflected on the learners in flipped learning, and homogeneity between experimental group and control group had not insignificant result, but flipped learning result for creative problem solving abilities was more effective than traditional classroom learning in software education.

VI. RESULTS AND SUGGESTIONS

In this study, we applied flipped learning to software education for students who are not majoring in computer science and examined its effects on their creative problem solving ability and learning satisfaction. As a result, the class with flipped learning was more effective in improving creative problem solving ability than the traditional lesson. In particular, it was effective in enhancing decision-making, one of the sub-variable of creative problem solving. This seems to be due to the fact that flipped learning based software education positively utilized problem solving learner activities.

Based on the results of this study, reexamining the software education curriculum for students not majoring in computer science is necessary. Designing a convergence software curriculum related to student major and employing theme-oriented software education in the flipped learning environment is highly recommended. Doing so will lead to a positive impact on fostering human resources in converged

software education.

REFERENCES

- [1] Korean Ministry of Education, *Software Operation Policy*, 2015.
- [2] K. S. Oh, “A study on the composition of computational thinking for programming education,” Doctoral thesis, Sungkyunkwan University, 2016.
- [3] Y. J. Kim, “Development and application of software education program for elementary school using ‘flip’ teaching method,” M.S. thesis, Seoul National University of Education, 2016.
- [4] M. Jung, “Effects of learning in flipped classroom on elementary school students’ learning, academic achievement, and attitude,” M.S. thesis, Korea National University of Education, 2014.
- [5] J. Y. Lee, Y. W. Kim, and Y. B. Kim, “Case study of learner-centered flipped learning,” *Educational Technology International*, vol. 30, no. 2, pp. 163-191, 2014.
- [6] H. Y. Lee, *2015 Teaching Guidebook Vol. 5*, Educational Competency Development Center of Hansung University, 2015.
- [7] D. Berrett, “How “flipping” the class can improve the lecture,” *The Chronicle of Higher Education*, vol. 58, no. 25, pp. 18-18, 2012.
- [8] E. J. Song and C. H. Lee, “Effect about achievement and motivation of flipped learning for science education at primary school,” 2016 Conference Journal, 297-301, Korean Science Education Society, 2016.
- [9] S. H. Kim, “Effects of learner-centered teaching and learning strategies for enhancing computational thinking,” *Journal of The Korean Association of Information Education*, vol. 19, no. 3, pp. 323-332, 2015.
- [10] Ministry of Science and Education, *Guideline and Reference Manual of Operation for Software Education*, Ministry of Science and Education, 2015.
- [11] J. H. Hwang, “Design of virtual reality learning space for enhancing creative problem solving ability,” M.S. Thesis, Korea National University of Education, 2010.



Ji Hyun Park was born on March 14, 1976 at Jeonju Korea. She is Ph.D. candidate at Chonbuk National University in 2015. Her major is education technology. She is interested in computer education, teaching method, cyber bullying.



Tae In Han was born on May 19, 1955 at Seoul, Korea. His major is AI of computer science. He is professor of Korea National Open University. He is interested in e-learning, deep and/or distance learning.