World Academy of Science, Engineering and Technology
International Journal of Economics and Management Engineering
Vol:8, No:7, 2014

The Impact of Training Method on Programming Learning Performance

Chechen Liao, Chin Yi Yang

***Abstract*—**Although several factors that affect learning to program have been identified over the years, there continues to be no indication of any consensus in understanding why some students learn to program easily and quickly while others have difficulty. Seldom have researchers considered the problem of how to help the students enhance the programming learning outcome. The research had been conducted at a high school in Taiwan. Students participating in the study consist of 330 tenth grade students enrolled in the Basic Computer Concepts course with the same instructor. Two types of training methods-instruction-oriented and exploration-oriented were conducted. The result of this research shows that the instruction-oriented training method has better learning performance than exploration-oriented training method.

***Keywords*—**Learning performance, programming learning, TDD, training method.

I. Introduction

STUDIES of programming can be generally divided into two categories, those with a software engineering perspective, and those with a psychological and educational perspective [1]. The interest here is in programming and initial development of individual programming learning.

Appropriate methods for training and programming development are mutually intricate and enduring matters. Approximately thirty years ago, technology education researchers commenced studying this topic with considerable enthusiasm. Two early works by Sackman, and Weinberg were significant in researching programming skill development from the psychological prospective. However viewed, programming language skills are fundamentally difficult for many learners to comprehend and master.

Recent research has focused on the motivation of students [2], in addition to student attributes on success in programming [3]-[5]. Although several factors that affect learning to program have been identified over the years, there continues to be no indication of any consensus in understanding why some students learn to program easily and quickly while others have difficulty. Seldom have researchers considered the problem of how to help the students to enhance the programming learning outcome.

Programming is the core in the computer course of seniors [1]. Nominal performance progress amongst many students on programming evaluations is an issue discussed at length with colleagues on H high school teaching council. Accepted agreements for improvement consequently draw attention to the assigning of more programming problems as homework, in addition to actively providing immediate and constructive advice. For the schools educators, attention is also needed for critical issues in regard to plagiarism, which necessities dealing with it in a resolute and unyielding manner.

Practice is one of the most important steps in learning the art of computer programming [6] and many researchers have explicitly noted that timely and appropriate assessment is fundamental to learning [7]. Unfortunately, evaluation of programming assignments is a tedious and error-prone undertaking, a problem compounded by commonly encountered large class sizes. As a consequence, students in such courses have a propensity to be assigned fewer programming assignments than should be idyllically completed. Given these institutional and class management limitations, the decision H high school has been to automate the homework grading process. This allows for students to electronically submit programming assignments and receive instantaneous feedback.

This method of on-line assessment, referred to as DICE, was established and initiated in February of 2005 [8]. At present, DICE system has been in operation for H high school's computer programming course for about three years, and over 2400 students have taken advantage of this evaluation method. DICE reached the initial goal of assigning more practice and evaluation immediately. DICE System makes available four levels of plagiarism detection to avoid an assortment of commonly encountered iniquitous acts, such as answer resending, adding white space, variable renaming, and semantic copying [8]. The consensus of the teaching council is that these unacceptable actions have been curtailed, with additional benefit in reaching the goal of providing students with immediate evaluation.

As anticipated, some well-known problems with the system have been encountered, which has caused a considerable number of users to be removed. Consequently, as it has been determined that this particular system should be upgraded to a more sophisticated evaluation method.

Due to the scarcity in the literature and the importance within the educational programming environment, there is a commitment here to better shed light on what attributes and attitudes students possess that leads to the ability to, without undo difficulty, learn programming, and how this understanding can aid the overall student population. Based on the research [9], we extend their finding to design different approach including exploration-oriented and instruction-oriented for programming learning students.

We refer the Test Driven Development concept to design

Chechen Liao and Chin Yi are with the Department of Information Management, National Chung Cheng University, Chiayi 621, Taiwan (e-mail: ccliao@ mis.ccu.edu.tw, vivian@ hkhs.tnc.edu.tw).

exploration-oriented and instructional-oriented training methods for students on our training system. For best improving learning outcome, we apply statistics by one-way ANOVA to find the impact of training method on learning performance. The results are the instructional-oriented training TDD-based training method benefits the learners on learning performance in programming learning.

II. Literature Review

A. Training Method

Training method is the primary treatment for individual learning style in this study. As the literature review of the learning style showed, adapting training method to accommodate key individual learning style led to improved performance.

Training materials deal with the organization of documentation provided to the training participants, whereas, training activities focus on the procedures followed in imparting training to the participants [9]. Training can be self-imposed or facilitated [9]. If facilitated, it can be conducted in a learning group or on one tutorial.

Another important dimension of training is the method employed. One major component of method is training approach [9]. Many training approaches are reviewed in the training literature, such as lecture, exploration-based, tutorial, and computer-assisted instruction.

Based on the previous researches, there are three main learning approaches, including instruction-oriented learning method, exploration-oriented learning method, and behavior modeling method. Instruction-oriented learning method has been characterized as the situation when "the entire content of what is to be learned is presented to the learner in final form" [10]. It is deductive, programmed, law learner control, compete material and feature control [11]. This approach teaches primarily by lecture and follows a deductive way to learning, where learners proceed from general rules to specific examples [12], [13]. Exploration-oriented learning method has been characteristic as "a matter of rearranging or transforming evidence in such a way that one is enabled to go beyond the evidence so reassembled to additional new insights" [14]. Glaser describes exploration learning as a process by which individuals are granted the freedom to impose their own structures on learning [14]. Exploration may also individual learns general concepts by starting with specific tasks or examples [15] and has been credited as the best way to learn personal computing [16]. It is inductive, trial and error, high learner control; incomplete learning materials and relevant task focus [11]. Behavior modeling method is a training process developed in the 1970s for building an individual's skill. This task-focused method involves a visual observation of the behaviors of a model performing task. The theoretical constructs of behavior modeling are well established [17].

Another component of the training method is the use of conceptual models. Conceptual models are depictions that contain the basic concepts of the target software [18]. The two types of conceptual models are analogical and abstract. Analogical models represent the target software in terms of another system. Abstract models are synthetic representations of the target software [9].

B. Test Driven Development

Test Driven Development (TDD) is a code development strategy that has been popularized by extreme programming. One always writes a test case before adding new code [19]. TDD can help to build software better and faster. It offers more than just simple validation of correctness, but also drives the design of a program. Some researchers report significantly increased code quality over traditional test-last approaches [20].

Computing and information technology educators have begun to call for the introduction of TDD into the curriculum [20]. Over the past five years, the idea of including software testing practices in programming assignments within the undergraduate computer science curriculum has grown from a fringe practice to a recurring theme [21].

C. Learning Outcome

Learning outcomes are the results of the training process [9]. There are principally two types of training outcomes: understanding (measured through learning performance) and motivation to use (measured through attitudes toward the system) [9].

In 1956, Benjamin Bloom developed a classification of levels of intellectual behavior important in learning. This became a taxonomy including three overlapping domains; the cognitive, psychomotor, and affective. Bostrom in 2006 draw from the research in educational psychology [22]-[25], he classified learning outcome into three categories: cognitive domain outcome, affective domain outcome and psychomotor outcome. Cognitive domain focuses on the mental awareness and judgment of the users [26]. Affective outcomes focus on the emotion aspects of the behavior. For the psychomotor domain, skill based outcomes focus on the ability to use the target system.

III. Research Model and Hypotheses

Referring the literature review, we establish the research model in Fig. 1. In this model, independent variable is training method and Dependent variable is learning performance.

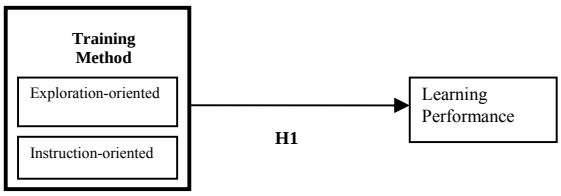

Fig. 1 Research Model

Many training formats are reviewed in the computer training literature, such as lecture, exploration-based, tutorial and computer-assisted instruction [13]. Nevertheless, little research has systematically examined the relative merits of training method in programming learning. However, Programming is

one of the main courses for computer science majors [1].

Exploration-oriented training method follows the traditional training method which was based on the ACM on the judge to train learner (http://acm.uva.es/problemset/). Under the traditional exploration-oriented training method, instructors just introduce problems to learners and let them solve by themselves. However, it shows the lack of logic system training in the programming approach. This method is designed for the learners who are excellent at programming, but it abandons many learners from programming as they don't know how to start to solve the problem.

As anticipated, some well-known problems in ACM with the system have been encountered, which has caused a considerable number of users to be removed. Consequently, it has been determined to follow the instruction method rule conducting TDD concept to guide learners to solve problems. The instructor needs to divide the problem into sub problem and let students conquer each sub problem. After they conquer every sub problem, then the problem will be solved finally. It's more intensive than exploration-oriented. From the discussing of the Teaching Council in H High School, they observed that most learners need intensive method in programming learning. According to the literature review of Jones [20], we have the following hypothesis:

H1. Participants in instruction-oriented (TDD) group will score significantly higher on learning performance measures than participants in the exploration-based (Non-TDD) group.

IV. Research Methodology

A. Variables and Measurements

Based on training method on Bostrom's study in 1990, we summarized there are two important components including training approach and conceptual of training method. Conceptual models are depictions that contain the basic concepts. The two types of conceptual models are analogical and abstract. Two main approaches found in the literature are exploration-oriented and instruction-oriented [11].

Instruction-oriented training approach is deductive, programmed, law learner control, compete material and feature control [11]. Exploration-oriented training approach is inductive, trial and error, high learner control; incomplete learning materials and relevant task focus [11].

For best learning with individual learning style in programming learning, we refer to the criteria of training method and the concept of Test Driven Development to develop a learning model. There is the introduction of the model we developed which named Test Driven Learning (TDL) in DICE system [8]. TDL model in DICE system can be coordinated to two dimensions, one for test cases in a teaching unit (XΩ) and the other for the test units in a test case (YΓ). The axis XΩ represents the coupling degree of test case sequences in a teaching unit. And the axis YΓ represents the coupling degree of test unit sequences in a test case. Obviously, XΩ represents more concepts in a teaching unit than YΓ, since YΓ represents more programming skills than XΩ. We divided the TDD plane in DICE system into sixteen areas by using four critical factors, denoted by ø for "none", > for "like, Γm for "modular" and ⊨ for "completed". For example, the XΩø - YΓø mode represents a teaching unit assignment with a "none" relationship between test cases, and so do "between" test units. The XΩ⊨ - YΓ⊨ mode means an assignment has TDD well-defined test cases with TDD-like test units. Most typical test-based automatic graders can support XΩ- YΓø modes by rearranging the problem sets. DICE system will support all sixteen modes.

In order to get the effectiveness as soon as possible, we roughly apply training approach with exploration-oriented and instruction-oriented to our subject to evaluate the learning outcome in every test unit. We define:

YΓøXΩø is exploration-oriented training approach and Non-TDD based training method also.

YΓ>XΩø are all instruction-oriented training approaches and TDD based TDD based training method also.

YΓmXΩø

YΓ⊨XΩø

Based on the cognitive domain outcome of Bloom [22], we define learning performance. Cognitive Domain focuses on the mental awareness and judgment of the user [26]. There are six major categories including knowledge, comprehensive, application, analysis, synthesis and evaluation starting from the simplest behavior to the most complex.

An examination will be referred the six criteria above and developed including objective questions for every session. There are six problems in the test referring from the ACM and were modified by the Teaching Council of H High School. The test will be held during the last 2 hours after training methods applying process. The range of possible scores for each test is between 0 and 100, with higher scores indicating better learning performance.

B. Participants

To obtain permission for conducting the experiment, the researcher contacted H High School in Tainan, Taiwan. We got the support from the administrators of H High School and all of the teachers in computer science education expressed their willingness to participate.

There are two experiments proceeding synchronously under the two teachers. In the A group there are 330 samples. Another group B has a total number of 176 samples. But, it was a pity for group a collective behavior in cheating happened in B group. Finally, we can't but surrender the data from group B. The remaining participates are the 330 learners in A group and no any missing data returned.

C. Training Materials and Procedures

There are 330 samples in the experiment taking the programming learning course in 10th graders of H High School. A series of lectures on the basics of programming skill concept including programming language introduction, basic input/output, decision making and repetition were delivered. It needs long time to progress the experiment for it must have function concept and the function session goes after all concept

of programming is settled down.

After establishing of function concept, we started to apply the training method to each group for one month. Finally, we had an examination to all participants and took the grades as their score of the course and also as the scale of learning performance.

D. Treatment Intervention: Training Method

We conducted a post-test only control group design and treatment was applied by random assigning to different training methods. Learners will be divided into two groups in terms of training method randomly. In our training process we separated every class into 2 groups by training method and learners in the two different training methods are parted to two isolated sections in the classroom which has 25 seats in each section.

Since Session 4, we applied to two different learning approaches on every learning style learning group. In the training process, what the instructor needs to do is to wait asking from learners and keep indifferently in each group of every class.

Two main training approaches in DICE system with TDL are the treatment. Referring the Test Driven Learning Model, Exploration-oriented training approach locates on the YΓ>XΩø area and instruction-oriented training approach locates on the YΓ>XΩø , YΓmXΩø and YΓ ⊧XΩø.

All training materials were referred to the ACM online judge guide, ACM which is the world's largest educational and scientific computing society, delivers resources that advance computing as a science and a profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources. An auto grading system - UVA online judge system which has collected many programming programs from the contests and thousands of computer science – oriented problems are provided to solve developed by ACM and maintained by University of de Valladolid in Spain. After registering an account, everyone in the world can download the problem sets and solve to judge by ACM UVA online judge system. There is a site ACM cat (http://luckycat.kshs.kh.edu.tw/) in Taiwan, they organized the problems from ACM UVA online judge site and classified the problem sets into different learners. For encouraging the learner to solve problem, everyone can translate the problem sets from ACM UVA online judge system and share to others.

In the exploration-oriented training method Non-TDD which is as designed as traditional ACM judge method. What the instructors needs to do is designing the problem, input and output data. All problems solving work will be reserved for the students. In the instructional-oriented training method TDD, the instructors need to deliberate on the problem and cut it into sub function to guide the student to accomplish the problem solving work.

V. Data Analysis

A. Descriptive Statistics

The majority of gender is male with 71.8% while the female has 28.2% (See Table I). Investigating the sample sizes from training method, the two training methods have the similar sample size. All participants need to choose the classification of Joint College Entrance Examination when entrancing the senior high school. We have much population in second classification with 47%, third/fourth classification department with 42.7%. The first classification of department only reaches 10% of total population.

TABLE I
DESCRIPTIVE STATISTICS OF SAMPLES

Item	Category	Freq.	%
Gender	Male	237	71.8%
	Female	93	28.2%
Training Method	Exploration–oriented (Non-TDD)	163	49.4%
	Instruction–oriented (TDD)	167	50.6%

Table II exhibits the average, standard deviation and the range of sore location on learning performance judging from training method. Instructional-oriented training method (TDD) has higher score Non-TDD in arithmetic mean.

TABLE II
DESCRIPTIVE STATISTICS OF TRAINING METHOD AND LEARNING PERFORMANCE

Training Method	Learning Performance			
	Mean	Standard Deviation	Min	Max
Non-TDD	14.25	19.79	0	80
TDD	21.51	27.14	0	100
Summary	17.92	24.03	0	100

B. ANOVA

The difference of means on learning performance between different training methods can be analyzed by one-way ANOVA. The test statistics of one-way ANOVA is 8.791, and its P-value under is <0.001 listing in Table III. The results show that TDD training method has significant better learning performance than non-TDD training method.

TABLE III
ONE-WAY ANOVA

	Mean Square	F	Sig.
Between Groups	2.634	8.791	<0.001
Within Groups	.300		

VI. Discussion and Conclusions

The main finding of this study is instructing learners gradually will benefit those most. Furthermore, adapting training method to accommodate key individual differences including learning style has led to improved performance.

We designed the auto grading system DICE with TDD concept after observing most learners will be abandoned in programming learning when they met the well-known program in ACM UVA online judge system which only has traditional training method: exploration-oriented training method conducted.

TDD is a code development strategy that has been popularized by extreme programming [19]. TDD is an

evolutionary approach to development which combines test-first development where writing a test before writing just enough production code to fulfill the test and refactoring. It offers more than just simple validation of correctness, but can also drive the design of a program faster. It is applicable on small projects with minimal training. It gives the programmer a great degree of confidence in the correctness of their code [21].

As TDD seems attractive, the idea of using TDD in the classroom is not revolutionary. Computing and information technology educators have begun to call for the introduction of TDD into the curriculum [20]. We also refer the TDD concept to design our DICE with exploration-oriented training method (Non-TDD) and instructional-oriented training method (TDD). Instructor can choose each training method selectable.

In our experiment, every participant applied each of the training methods with DICE by randomly assignment. The result from data analysis shows instructional-oriented training method (TDD) benefits most of learner on learning performance in programming learning. From the observation of training process, we found learners in instructional-oriented training method (TDD) more enjoy discussing with their mates than Non-TDD group. As to Non-TDD group when meeting puzzles, they tend to rely on the direction of the instructor.

REFERENCES

[1] A. Robins, J. Rountree, and N. Rountree "Learning and teaching programming: a review and discussion," Computer Science Education, (33-2), 2003, pp. 137-172.
[2] J. Tony "The Motivation of students of programming," ACM SIGCSE, (33:3), 2001, pp. 53-56.
[3] A. Goold, and R. Russell "Factors affecting performance in first-year computing," SIGCSE Bull., (32:2), 2000, pp. 39-43.
[4] P. Byrne, and G. Lyons "The effect of student attributes on success in programming," ITiCSE: Proceedings of the 6th annual conference on Innovation and technology in computer science. ACM press, 2001, pp. 49-52.
[5] B. Susan, and R. Ronan "Programming: factors that influence success," Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, 2005.
[6] D. Christopher, L. David, and O. James "Automatic test-based assessment of programming: areview,"ACM Journal of Educational Resources in Computing, (5:3), September 2005, Article 4.
[7] L. David, and O. James "Automatic test-based assessment of programming: a review," ACM Journal of Educational Resources in Computing, (5:3), September 2005, Article 4.
[8] L. R. Chien, D. J. Buehrer, and C. Y. Yang, "DICE, a Parse-Tree based on-line assessment system for a programming language course," The Third Conference on Computer and Network Technology, 2007.
[9] R.P. Bostrom, L. Olfman, and M. K. Sein, "The importance of learning style in end-user training," MIS Quarterly, (14:1), 1990, pp. 101-119
[10] D.P. Ausubel, The psychology of meaningful verbal learning, New York, Grune and Stratton, 1963.
[11] S.D. Davis, "Training novice users of computer systems: the roles of the computer interface," Doctoral Dissertation, Indiana University Bloomingtom, IN, 1989.
[12] H.W. Chou, and T.B. Wang "The influence of learning style and training method on self-efficacy and learning outcome in WWW homepage design training," International Journal of Information Management, (20:6), December 2000, pp.455-472.
[13] H. W. Chou, "Influences of cognitive style and training method on training effectiveness," Computers and Education, (37:1), 2001, pp. 11-25.
[14] J. Bruner, Toward a Theory of Instruction, New York: W. W. Norton, 1966.
[15] H. Taba, "Learning by discovery: psychological and educational rationale," The Elementary School Journal, (63:6), 1963, pp. 308-316.
[16] J. Brynda, "End user training: lend me your ear, I will teach you a PC package," Computing Canada, (18:1), 1992, pp. 48.
[17] A. P. Goldstein, and M. A. Sorcher, Changing Supervisor Behavior, New York: Pergamon, 1974.
[18] D. A. Norman, "Some observations on mental models" in Mental Models, A.L. Stevens and D. Genter (ends.), Lawrence Kawrence Erlbaum Associates, Hillsdale, NJ, 1983, pp. 7-14.
[19] K. Beck, Test Driven Development: By Example, Addison-Wesley, 2003
[20] C.G. Jones, "Test-driven development goes to school," Journal of Computing Sciences in Colleges, Vol. 20, 2004, pp. 220-231.
[21] S. H. Edwards, "Teaching software testing: automatic grading meets test-first coding." In Proceedings of the OOPSLA'03 Conference. Poster presentation, 2003, 318-319.
[22] B. S. Bloom, M. D. Englehart, E.J. Furst, W.H. Hill, and D. R. Krathwohl, Taxonomy of Educational Objectives: The Claasification of Educational Goal. Handbook 1: Cognitive Domain. New Work: MaKay, 1956.
[23] D.R. Krathwohl, B. S. Bloom, and B. B. Masia, Taxonomy of educational objectives: The Classification of Educational Goals. Handbook 2: Affective Domain. New York: McKay, 1964.
[24] E.J. Simpson, "The Classification of Educational Objectives, Psychomotor Domain," Illinois Teacher Home Economics, 1966, 10, pp. 110-144.
[25] B. S. Bloom, Taxonomy of Educational Objectives: the Classification of Educational Goals. Longman, New York, 1984.
[26] S. Gupta, and R.P. Bostrom "End-user training methods: what we know, need to know," Proceedings of 2006 SIGMIS CPR Conference, 2006, pp. 172-182.