DIGITALES ARCHIV

Kazanidis, Ioannis; Moussiades, Lefteris

Article

The use of pair programming in the object oriented programming instruction

MIBES transactions

Provided in Cooperation with:
Technological Educational Institute (TEI), Thessaly

Reference: Kazanidis, Ioannis/Moussiades, Lefteris (2017). The use of pair programming in the object oriented programming instruction. In: MIBES transactions 11 (1), S. 40 - 47.

This Version is available at:
http://hdl.handle.net/11159/1887

Mitglied der

ZBW Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Leibniz-Gemeinschaft

The use of Pair Programming in the Object Oriented Programming Instruction

Ioannis Kazanidis, Lefteris Moussiades
Computer and Informatics Engineering Department
Eastern Macedonia & Thrace Institute of Technology
{kazanidis, lmous}@teiemt.gr

Abstract

Pair programming is an example of collaboration that is increasingly being used in programming training, both for the benefits of collaborative learning and for the programming experience that is closer to the real-world software development. International surveys reveal the usefulness of the method and the benefits it can bring to learners. In this paper, we present a case study in which 51 students, who have attended a laboratory course of object-oriented programming in a higher education institute, were divided into two groups (control group and experimental group) and their academic performance was measured. Initial results revealed that students who worked in pairs developed better code and had a slight higher academic performance from the students who work individually, they were very satisfied with the process they followed and they asked for the application of pair programming to other programming courses.

Keywords: pair programming, collaborative learning, object-oriented programming

Introduction

Programming instruction is one of the most important issues faced by a programming educator. The challenges in teaching a programming subject have been widely recognized by the educational community (McChesney, 2016). In particular, the teaching of object-oriented programming presents a great deal of difficulty and as a result, alternative teaching methods have been proposed(Xinogalos, 2014).

The method of collaborative learning was one of the first that was introduced to IT courses. Collaborative learning promotes the exchange of ideas among the members of a group to increase trainees' interest, active participation, cultivation of social skills and the promotion of critical thinking (Gokhale, 1995). Inevitably, collaborative learning was also applied to computer programming. One of its applications is the pair or collaborative programming in which two learners work together in a computing system.

In this paper, we present a case study in which pair programming was applied to a sample of 51 students divided into two groups (control group and experimental group), in an object-oriented course of study in tertiary education. The initial results showed a slight improvement in the performance of students who worked in pairs while they were very satisfied with the context they followed and they would like to apply pair programming to other programming courses as well.

At the next sections of the paper pair programming method and its main characteristics are presented in more details, the related literature is analyzed, and the case study at Eastern Macedonia and Thrace

Institute of Technology is presented along with the qualitative and quantitative results that arise. The work concludes with the findings from the experiment and the proposed future applications.

Pair Programming

Collaborative programming has been successfully implemented in many IT courses and multiple benefits have been recorded such as the development of social skills and co-operation which are benefits encountered in the real-world programming (Falkner et al., 2013; Sanjay & Vanshi, 2010), the sharing of knowledge among the trainees (Falkner et al., 2013), the enhancement of learning through the interaction of developers (Tzombanoudi & Satatzemi, 2012), greater sense of responsibility (Porter & Simon, 2013) and self-perception (McChesney, 2016), the better performance in evaluation (Hanks et al., 2011; Salleh, Mendes & Grundy, 2011), the highest quality code with fewer errors (Sanjay & Vanshi, 2010; Zacharis, 2009). Trainees working in pairs noted that in the initial phases of programming they analysed and discussed the design of the program in contrast to their colleagues who were working individually and tend to postpone actions that required critical thinking (Li & Kraemer, 2014).

However, some problems have been reported in the implementation of pair programming (Hanks et al., 2011; Salleh, Mendes & Grundy, 2011) such as the large differences in the contribution of each member of the group, the feeling of the students that valuable time is lost, (McChesney, 2016), and the more time required for team members coordination (Duque & Bravo, 2008).

Case Study

In this paper, we describe the application of pair programming to a course of the 3rd semester of the Eastern Macedonia and Thrace Educational Institute of Technology (EMT TEI for short) entitled "Object Oriented Programming". The experiment aims to study the potential benefits of collaborative programming in higher education and in particular in the course of object-oriented programming, as well as to capture students' perceptions and preferences to improve the teaching in the classroom. The main research challenge is whether the pair programming approach can impact learning effectiveness. The latter can be determined by measuring students academic performance and training satisfaction with the following research questions (RQs): RQ1: Does the pair planning approach has a positive impact on students academic performance in Object-Oriented Programming Course?

RQ2: Do students have better learning experience through pair programming approach?

Methodology

Participants

A total of 51 third-semester students (38 males and 13 females) of the Computer and Informatics Engineering Department of EMT TEI participated voluntarily in this study. Despite the female participants were significantly less than their male counterparts, the sample was regarded as acceptable, since this gender ratio is typical

in Greek ICT-focused higher education departments and a balanced sample would be either too small or unattainable.

The students were divided into two main groups. The control group (CG), students worked individually, and the experimental group (EG) where students worked in pairs using the same computing system.

Regarding the formation of the pair programming teams, according to the relevant literature, the creation of a team is made either based on the level of knowledge of the learners or by random combinations or by allowing the trainees to choose their partners. Since the research has not yet resulted in the best way for partner selection (Sanjay & Vanshi, 2010; Zacharis, 2009), we ended up letting the students choose their partner themselves and examine the reasons that led them to their choice.

Research Tools

The research tools used were one posttest, one opinion mining questionnaires and a discussion with the focus group.

The posttest was intended to evaluate students' academic performance. It consisted of 3 multiple choice questions and one open-ended question.

The second questionnaire was given only to the participants in the experimental group, and it was designed to study the learners' satisfaction with the followed approach. It was consisted by 10 multiple choice questions (five of them were answered on a 5-point Likert-type scale ranging from strongly disagree (1) to strongly agree (5)) and one question for programming development.

The Focus Group is a qualitative research method, aims either to confirm or not the findings of quantitative research through the questionnaire and to study in more depth and to justify the actions and the results produced through the quantitative evaluation method with the questionnaires

One additional assessment tool was the program that students develop during the experiment. Studying the quality of the program, the percentage of completion and possible error patterns may provide researchers with useful outcomes

Procedure

One week before the experiment, we informed students about what was going to happen in the next lesson and divided them into the two main groups. The students of the experimental group asked to choose their partner during the week before the experiment.

On the day of the experiment, we had two teaching hours available. Since the real time was 90 minutes, it was distributed as follows. During the first 10 minutes, the teacher briefed on the process that would be followed. In the next 50 minutes, students should solve the required exercises by working either individually or in pairs. Students working in pairs were asked to exchange driver and navigator roles after the first 25 minutes, to have equal time in each role. Students were then asked to store their programs and send them to the teacher for evaluation. The next 10 minutes, students were required to

answer the posttest. In the last 20 minutes, only the participants in the experimental group stayed in the class and answered the satisfaction questionnaire for 10 minutes while the last 10 minutes used for discussion on the students' pair programming experience.

In particular, students were asked to work either in groups of two or individually and solve, in two class hours, the following problems programming in C ++.

Exercise 1: Design a class for students that stores the student's name, three courses, and three grades as well as the appropriate methods.

Exercise 2: Implement a method in the following program that reads the three student grades and finds the average grade.

Exercise 3: What changes have to be done in the given code to store the student grades in a table that is available throughout the class? What problem - malfunction exists in the given example? Make any corrections that you think as appropriate.

The content of the exercises was covering the three phases that according to Boehm (1981) correspond the 59% - 68% of a program developing cost i.e. design, development and evaluation. Students were asked not to compile before completing their code, according to the programming process proposed by Humphrey (1995).

Results

Fifty-one (51) individuals were involved in the experiment, 25 of them in the control group (males n=19, females n=6), and 26 in the experimental group (males n=19, females n=7)as shown in Fig.1.

To study whether the first research question was valid, we conducted two t-tests. The first t-test examined whether are any differences in the student's scores regarding the program they developed during the experiment. The second T-test checked whether students of the students of the two groups had different performance on the posttest.
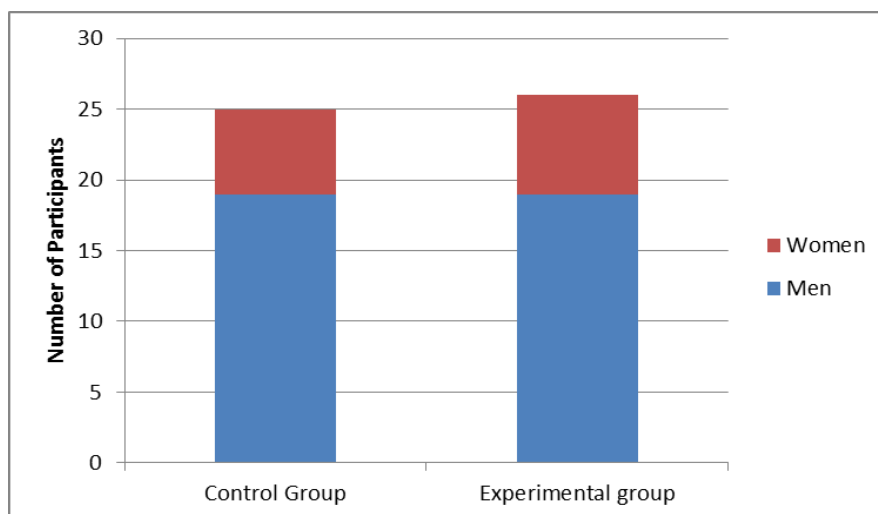

Figure 1: Participants per group and gender

Programming assignment

Before employing T-test, Levene's homogeneity test was conducted (Levene, 1960). The test showed that there was equality at 5% (p = 0.336> 0.05), which indicated that the homogeneity test had not achieved statistical significance; therefore, t-test could be applied. The results of the T-test (Table 1), show that there is a significant statistical difference at the level of 5% (p = 0.048> 0.05) between the scores of the students who worked in pairs (mean = 5.961) and the students who worked individually (mean = 5.020). This difference indicates that students working in pairs produce better code than those who programmed individually.

Posttest results

The posttest examined student academic performance. A t-test was conducted to examine if there is any statistically significant difference in the students scores on the posttest. A Levene's homogeneity test was conducted, prior to the t-test, to assess the equality of variances. The Levene's test show that the F value was equal to 0.049 (p=0.825>0.05) which indicated that the homogeneity test had not achieved statistical significance and therefore t-test could be applied. The t-test that took place right afterwards show that students that work in pairs had better academic performance (Mean score=6.288) comparing to the students that work individually (Mean score=5.4). However, this difference is not statistically significant since p=0.063 (Table 1) which is higher than the threshold of 5%. This result shows that students working in pairs may succeed slight better academic performance than their fellows even if it is not statistically significant.

Table 1: Overall scores in program development and posttest

Scores	Groups	N	Mean	SD	t	df.	Sig.
Program score	CG	25	5.020	1.544	2.027	49	0.048
	EG	26	5.961	1.760			
Posttest score	CG	25	5.400	1.626	1.903	49	0.063
	EG	26	6.288	1.703			

Satisfaction questionnaire results

Table 2 provides descriptive statistics including means and standard deviation for the students satisfaction regarding the pair programming method followed from EG of students. The satisfaction questionnaire answered by the members of the EG and revealed that students of EG were satisfied with the process. Given that the full agreement score was 5 and complete disagreement 1 the students stated they enjoyed the collaborative programming process (3.461).

The next four questions, examine the acceptance of this method by the students according to the Technology Acceptance Model - TAM (Davis, 1989) which examines the perceived ease of use and usefulness of a technology. The students stated that they easily co-operated with their colleague (3.576) and agreed to a greater extent than all the other questions (3.807) that they consider pair programming as a useful method for programming courses, and they would like to program in pairs in more courses (3.615).

Table 1: Training satisfaction questionnaire results

Question	Mean	Std. Deviation
1. I enjoyed the pair programming process	3,461	0,947
2. It was easy for me to co-operate with my colleague	3,576	0,808
3. I consider useful the pair programming method	3,807	0,980
4. I would like the pair programming method to be used in more courses	3,615	0,982
5. I believe that I learned more by working in pairs than individually	3,461	0,811

Another one thing that was examined by the questionnaire was how the students choose their partner. Given the importance of the decision regarding the development of the program, we wanted to see which the criteria are for the partners selection. According to the students statements, 50% of them chose their partner mainly of their friendship, 19% due to his/her knowledge of the domain, 23% randomly (they were at the time of the workshop or they had not found a group) and 8% state another reason. It seems in this case, that friendship with a person is the most critical factor of partner choice. In addition, the students stated that, what they liked most in the pair programming approach was that talking to someone on the exercise made things easier for them (42%), while the thing they liked less was the difference in perceptions with their partner and the proposed solutions in some issues (38%). In the question "Propose ways to change or improve the whole process and experience," two students stated that the whole process could be further improved by exercises that require more critical thinking so that more analysis and discussion take place between the students.

Focus group

To confirm the above quantitative data, a focus group discussion was held for 10 minutes. Initially, students were asked to express their general view about the process, and later a discussion in more depth about the questions of the satisfaction questionnaire took place, to see if the results were verified and which factors led students to give the specific answers.

Overall, the students view was very positive about the whole process. The only point they focused on, was the coordination of the two partners and the time they spent for this. They stated that they would develop more code and faster if the two partners were working simultaneously on different computers and different tasks on the same program; however, they agreed that within education context the adopted process was very useful.

To the question, whether they feel that pair programming helped them to learn better than programming individually, they agreed that pair programming has a positive impact and that the exchange of ideas and knowledge works positively to the learning of the domain. They also stated that the learning experience was better than the traditional programming courses where they are working in the lab individually and

would like to see more programming courses of their department to adopt this method.

Despite the above positive evidence, there were three students who stated that they had difficulties working with their partners. Indeed, two of them reported that there was a difference in the knowledge of the group members, which resulted to take control only one student and the other watching passively. Finally, a student stated that it would be useful if it was possible for the partners to process the code at the same time, keeping their roles, and the observer can easily point, mark or insert notes into the program.

Discussion and Conclusion

This paper presents an application of pair programming in an object-oriented programming course. The paper aims to examine whether the students who programmed in pairs had better academic performance and learning experience. In the experiment that conducted, both quantitative and qualitative data was acquired. Quantitative analysis verified part of the first research question since students working in pairs achieved a slightly better average score from the students who worked individually with not any significant statistical difference. Nevertheless, this particular result shows that there is a positive impact of the pair programming method on learner performance. In addition, the learning experience according to the majority of participants was very positive, and it seems that they prefer programming in pairs by individual programming in education settings.

These results come to verify previous studies (Hanks et al., 2011; Salleh, Mendes & Grundy, 2011; Sanjay & Vanshi, 2010; Zacharis, 2009) and demonstrate the usefulness of the method in higher education. Additionally, the suggestion that was stated for concurrent code processing from both students leads to the consideration of pair programming solutions using specific systems such as that proposed by Tsompanoudi et al. (2015) for in-class.

This study has the following limitations:
a) the small sample of students used in this study. Specifically, the students who worked in pairs were 26 and only 13 teams were created.
b) all participants in the sample were students of a particular higher education institution with similar characteristics. In this sense, it is difficult to generalize the results.
c) The short duration of the experiment set at the two teaching hours does not allow the safe prediction of participants behaviour. Their performance or opinion might be changed on specific issues if the experiment was taking place for more time during the semester.

Due to the above limitations, we plan to evaluate the pair programming method in a whole semester time using a larger sample of students. However, at the new experiment, we will focus on the learning mechanisms through the programming method to understand which are the factors that enable learners to get better performance, how they work and how to further improve the teaching of object-oriented programming through students' collaboration.

References

Boehm, B.W., (1981), "Software Engineering Economics," Prentice-Hall, Englewood Cliffs

Davis, F.D., (1989), "Perceived usefulness, perceived ease of use, and user acceptance of information technology," MIS Quarterly (13), 319–339.

Duque, R. & Bravo, C., (2008), "Analyzing Work Productivity and Program Quality in Collaborative Programming," In Proceedings of the 3rd International Conference on Software Engineering Advances (ICSEA '08), IEEE Computer Society, Washington, DC, USA, 270-276.

Falkner, K., Falkner, N. & Vivian, R., (2013), "Collaborative Learning and Anxiety: A phenomenographic study of collaborative learning activities," *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE),* 227-232.

Gokhale, A., (1995), "Collaborative learning enhances critical thinking," *Journal of Techology Education*, **7**(1).

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L. and Zander, C., (2011), "Pair programming in education: a literature review," Computer Science Education, **21**(2), 135-173.

Humphrey, W.S., (1995), "A Discipline for Software Engineering," Addison-Wesley, Reading.

Levene, H., (1960), "Robust tests for equality of variances," In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling, I. Olkin et al. (Eds.) Stanford University Press, 278-292.

Li, Z. & Kraemer, E., (2014), "Social effects of pair programming mitigate impact of bounded rationality," *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 385-390.

McChesney, I., (2016), "Three Years of Student Pair Programming – Action Research Insights and Outcomes," *Proceedings of the 47th ACM Technical Symposium on Computer Science Education*, 84-89.

Porter, L. & Simon, B., (2013), "Retaining Nearly One-Third more Majors with a Trio of Instructional Best Practices in CS1," *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 165-170.

Salleh, N., Mendes, E. & Grundy, J., (2011), "Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review," *IEEE Trans. Softw. Eng*, **37**(4), 509-525.

Sanjay, G. & Vanshi, K., (2010), "A Novel Approach for Collaborative Pair Programming," *Journal of Information Technology Education*, USA, **9**, 183-196

Williams, L., Kessler, R., Cunningham, W. and Jeffries, R., (2000), "Strengthening the Case for Pair Programming," *IEEE Softw.,* **17**(4), 19-25.

Xinogalos, S., (2014), "Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy," *Education and Information Technologies*, 1-30.

Zacharis, N., (2009), "Evaluating the Effects of Virtual Pair Programming on Students Achievement and Satisfaction," *International Journal of Emerging Technologies in Learning*, **4**(3), 34-39

Tsompanoudi, D., Satratzemi, M. & Xinogalos, S., (2015), "Distributed Pair Programming Using Collaboration Scripts: An Educational System and Initial Results," *Informatics in Education*, 14(2), 291-314

Tsompanoudi, D. & Satratzemi, M., (2012), "Cooperative Distance Programming and Adaptive Support for Collaboration," In Proc. of the 6th Panhellenic Conference Didactics of Informatics, 73-82, Florina, Greece.