Volume 14, No. 3
May 2016
ISSN: 1545-679X

Information Systems Education Journal

In this issue:

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published reviewed published by **ISCAP,** Information Systems and Computing Academic Professionals. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCon (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the 1st Edition of Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

Information Systems Education Journal

Evaluating Students' Perception of Group Work for Mobile Application Development Learning, Productivity, Enjoyment and Confidence in Quality

Loreen M. Powell
lpowell@bloomu.edu
Department of Business Education and Information & Technology Management
Bloomsburg University
Bloomsburg, PA 17815, USA

Hayden Wimmer
hwimmer@georgiasouthern.edu
Department of Information Technology
Georgia Southern University
Statesboro, GA 30260, USA

Abstract

Teaching programming and mobile application development concepts can be challenging for instructors; however, teaching an interdisciplinary class with varied skill levels amplifies this challenge. To encompass a broad range of students, many instructors have sought to improve their lessons and methods by experimenting with group/team programming. However, these studies focused on the instructor's usage of the method and not the students' perceptions of the method. This study was conducted to understand students' perceptions regarding the effectiveness of the student's group/team experience and learning outcomes when developing a mobile application. Results were favorable towards using group work for mobile application development learning, productivity, enjoyment and confidence of quality.

Keywords: programming, mobile application development, group work, team-based learning, collaborative learning,

1. INTRODUCTION

Many employers want technology savvy students that can collaborate with others nationally and globally. As a result, non-technology degree students are enrolling into technology based courses, including a computer applications/programming course to secure an edge in the job market. Thus, an applications/programming course may be considered interdisciplinary in nature.

Teaching programing and mobile application development concepts can be challenging for instructors; however, teaching an interdisciplinary class with varied skill levels amplifies the challenge. In an effort to find a solution to these challenges, many instructors have experimented with different collaborative learning techniques or software (Medina, Gomez-Perez, Neito-Reyes & Santos, 2013; Faja, 2014) or group/team learning methods.

There have been several similar studies that also found that students enjoy working in teams (Williams & Kessler, 2001; Cliburn, 2003; McDowell et al., 2006; Howard, 2007; Chigona & Pollock, 2008; Mentz at al., 2008; Zacharis, 2011). However, a study has not been found regarding students' perceived effectiveness of using group/team work for mobile application development in a hands-on programming environment. This exploratory study surveyed students to understand their perceptions of using group work for mobile application development learning, productivity, enjoyment, and confidence in quality. This work has practical implications for programming faculty and practitioners alike. The remainder of this paper is structured as follows: a brief review of programming pedagogy and collaborative learning (group/teamwork), the importance of engaging students through mobile application development, Stencyl, method, results, conclusions and limitations.

2. LITERATURE REVIEW

For years, it has been a challenge for students to learn programming skills (Sleeman, 1986; Ebrahimi, 1994; Faja, 21014; Jenkins 2002; Kinnunen et al. 2007; Mow, 2008; Nikula, Gotel, & Kasurinen, 2011; Powell & Wimmer; 2015). Babb et al. (2014) defined several known pedagogy failure mechanisms for students learning programming skills. One of pedagogy failure mechanisms reported was the lack of appropriate team/group work formats which support collaborative and peer driven learning.

Collaborative learning is "when a small group of students work together to complete an academic task" (Chinn & Chinn, 2009). Previous research has identified collaborative learning as a good instructional tool in higher education (Baer, 2003).

Michaelsen, Knight, and Fink (2004) expanded upon collaborative learning and developed a team-based learning (TBL) technique. Their technique TBL stresses the importance of using small groups to help apply key. TBL techniques has been used in the medical, engineering, business, sciences, technology, and liberal arts courses.

Lasserre (2009) adapted the TBL technique for a first semester programming class. She reported that student drop rates decreased as a result of using the TBL technique within her course. Lasserre and Sztostack (2011) further reported additional increases in grades as a result of TBL. A more current research study by Faja (2014) utilized conducted research on the use of paired programming for students. He defined paired programming as a collaborative learning technique that involves two students working together, side by side, sharing a computer to complete an academic task.

Faja (2014) examined students' perceptions on effectiveness of pair programming. He utilized a survey adopted from Chigona and Pollock (2008) and Howard (2007) to collect data from introductory computer programming classes. His results indicated that students perceived paired programming beneficial in learning and they also enjoyed paired programming.

Hu and Shepard (2014) utilized the process oriented guided inquiry learning (POGIL) to help teach first year programming students. POGIL is similar to TBL in that it uses student teams. However, POGIL is said to be better focused on concepts and process skills development. This study found that students who worked in teams using the POGIL method experienced increased grades.

Previous and current studies on group work/TBL in the classroom tend to focus on the typical programing languages, including, but not limited to, Visual Basic, Java, and C++ (Lasserre, 2009; McKeown, 2004,). There are few studies that focus on group work using mobile application development software.

A recent paper by Hoffman (2014) explains an interdisciplinary group approach for a game design, mobile web and application development course. Students utilized App Inventor for their mobile application development group project. It was found that some problems occurred within groups, in that, group members were delinquent or missing their parts. As a result, other group members had to pick up their work. It was also reported that planning and delegating issues occurred as a result of an open-ended project. However, the paper does not provide data on student perceptions regarding the usage of groups for mobile application development.

Importance of Engaging Students through Mobile Applications

Today, with the presence of advanced technologies and the extended availability of the smart mobile technology devices, many educators are exploring ways to enhance students' learning (Burd, Barros, Johnson, Kurkovsky, Rosenbloom & Tillman, 2012; Ching-Chiu Chao, 2006; Klopfer, 2008). While many educators may think that mobile technology is just another trend in the evolution of technology, smart mobile technology has morphed into much more than the next stage of the computer revolution partially because of its associated cost and student acceptance (Burd et al., 2012). Almost every incoming college or university student carries a smart mobile technology device. Madden, Lenhart, Duggan, Cortesi, and Gasser (2013) conducted a nationally representative phone survey study regarding smartphone adoption among American teens (ages 12-17). Their results stated that 78% of teens have a mobile device. More alarming, they reported that one in four teens are "cell-mostly" internet users. Cell-mostly users are defined as those who only use their phone to access the internet.

Another research study reported by Smith (2013) indicated that 91% of the adult population has a mobile phone/device. More specifically, he reported that 79% of college aged students (18-24) have a smartphone. To further explain the impact and importance of mobile devices, Smith (2010) stated that one in five individuals claim they would rather spend a week shoeless than a week without their mobile phone. Hall (2013) also believes that teens are obsessed with smartphones. He classifies teens as having a "mobile first" mentality to the Internet similar to Madden et al.'s "cell-mostly" Internet users.

Given the ubiquity of smart mobile technology devices and our social attachment to them, it is essential to engage students within a programming classroom via mobile application development. Today, mobile software creation applications such as Stencyl can be used to further apply students programming knowledge.

Stencyl (www.stencyl.com)
Stencyl is a downloadable application that is available free and in a paid version form to create mobile applications on your personal computer (PC), or Mac computer. Stencyl also has a jigsaw-puzzle piece graphical interface (GI) that has been very successful in previously developed programming applications such as Scratch (www.scratch.mit.edu), Turtle Logo (http://logo.codeplex.com/), Alice (www.alice.org), and App Inventor, (http://appinventor.mit.edu/). These applications focus on logic (Burd et al., 2012).

There is a limited amount of research conducted on the use of Stencly in the classroom. Most of the existing research has focused on programming or usability issues.

3. METHOD

The purpose of this research study is to understand the student's perceived value of using group work in hands-on applications/programming class to develop a mobile application. The research questions are:

1. In a hands-on programming course, how will students perceive group work when developing a mobile application?

2. In a hands-on programming course, how will students perceive the four category outcomes (perceived quality, perceived productivity, perceived learning and enjoyment) from using group work to develop a mobile application?

3. Will there be any significant difference between students mean scores among of the four category outcomes (perceived quality, perceived productivity, perceived learning and enjoyment) from using group work to develop a mobile application?

4. Will there be a significant difference between the gender perceptions in using group work to develop a mobile application?

Subjects were undergraduate students enrolled in a medium sized 4-year state institution. Students were enrolled in a traditional face to face section of an applications/programming course where students learn to program with Scratch, Visual Basic, and Stencyl. The purpose of this course is to present solutions for the business environment using Object Oriented Language (OOL) and other web-based development tools. The primary goal of the course is on programming. Students learn how to program within visual basic and other web based mobile application development tools such as Stencly. Students also learn how to develop usable applications including mobile applications. Approximately 75% of the course focuses on programming and the other 25% of the course focuses on how to design, develop, and work with applications.

Over a 14-week semester, the course consisted of three fifty minute classes per week (Monday, Wednesday and Friday). The class was a traditional face to face course held in a computer lab for a hands-on learning experience. The class was structured so that the first 3 weeks, students learned/worked with introductory programming concepts and Scratch.com. The following 8 weeks, students learned/worked with Visual Basic. Finally, the last 3 weeks' students learned/worked with Stencyl.

For the first 11 weeks, the instructor followed an "introduce, reinforce, apply, and assess" format. To introduce the concepts, the instructor held a lecture style PowerPoint session to go over key concepts for each chapter. To reinforce the key concepts learned, the instructor illustrated hands-on step by step ways to code for each chapter. To apply the key concepts learned, the instructor worked with the students by illustrating and guiding them in application development and programming. Finally, to assess the key concepts, the instructor gave a theory and a hands-on assessment. Each assessment was graded and distributed back to them within one week. An entire class period was spent reviewing each exam.

Throughout the 11 weeks, students learned basic programming concepts using Scratch and Visual Basic. Topics discussed were:

- Introduction to programming
- Program and Graphical User Interface Design
- Program Design and Coding
- Comments
- Variables and the Arithmetic Operations
- Decision Structures
- Loop Structures
- Using Procedures and Exception Handling
- Using Arrays and File Handling

The last 3 weeks of class, students worked with Stencly. Stencyl was placed towards the end of the semester because the students needed to learn the basic concepts before working with Stencyl. The instructor charged students with the task of working with in groups to create a mobile application using Stencyl. The students self-selected their groups.

The mobile application assignment was specifically left open-ended for the students to use creativity in their development process. The only graded requirements were that the application must have at least 3 different levels, 3 different objects and controls, as well as be classroom appropriate. Students were also required to create a story board of their mobile application.

Over the course of three weeks, students work with the instructor and their classmates to share ideas and build their mobile application. Each group briefly described their mobile application and then randomly challenged a student from a different group to come to the front of the room and try to use their mobile application.

Data was collected at the end of the semester via an IRB approved survey. The survey was adopted by Faja (2014) and modified to specifically address using group work to develop a mobile application. It is important to mention that Faja's (2014) survey was adopted from Chigona and Pollock (2008) and Howard (2007). Hence, this research survey was also adopted from the same researchers.

Our survey contained 12 questions/statements. The first two questions/statements were demographic in nature. The remaining 10 questions/statements were aimed at gathering information from the students regarding their perceptions of using group work for mobile application development. The survey used a Likert scale with response categories of Strongly Agree (5), Agree (4), Neither Agree nor Disagree (3), Disagree (2), and Strongly Disagree (1).

The survey was optional; students were not required to complete the survey. The instructor of the course was not present when the survey was electronically administered by another faculty member. The survey was anonymously completed by the students.

4. RESULTS

Statistical analyses were conducted using the Statistical Product and Service Solutions (SPSS) software. Various statistical test were used in this study. Specifically, a Cronbach's alpha analysis was used to test the reliability of the data set. Descriptive statistics were used to summarize the demographic data regarding the students. Also both descriptive and inferential statistic, including mean and standard deviation were used as a measure of central tendency and spread of the data set. Finally, paired t-tests, and two-tailed independent t-tests were used to test the research questions.

Reliability Testing

Reliability testing is typically used in survey instruments with summated and multi-point scales. The Cronbach's Alpha, which measures the internal consistency, is the most popular test for assessing reliability (Santos, 1999). When using the Cronbach's Alpha for testing reliability, "alpha coefficient ranges in value from 0 to 1 (Santos, 1999)." The typical acceptable Alpha reliability threshold is 0.7. Hence, the higher the Apha score, the better the reliability (Nunnaly, 1978; Santos, 1999). Reliability testing was conducted on the survey instrument. The Cronbach's Alpha was .946. Hence, this shows a good internal reliability because it is above the acceptable threshold score.

Descriptive Statistics for the Student Population

The overall sample size included 33 undergraduate business students enrolled in an undergraduate applications/programming course which is taught as part the Information and Technology Management (ITM) curriculum.

There were a total of eight different student groups within the course. The size of the groups ranged from three students to six student members. Specifically, there were three groups consisting of three student members, four groups consisting of four student members and one group consisting of six student members.

It is important to note that Institutional Research Board (IRB) approval required the survey to be anonymous and not mandatory for students. Therefore, collecting demographic information such as year of study and the discipline/major was not permitted. As a result, demographic data shown in Tables 1.1 and 1.2 was not collected via the survey. This data was retrieved from the university's student enrollment system and reported as a whole.

Table 1.1 shows the overall composition of the entire class with regards to their year of study. The data shows that there are few freshmen enrolled in the course and that majority of students are juniors or seniors.

	Frequency	Percent
Senior	4	15.10%
Junior	10	42.40%
Sophomore	14	30.30%
Freshmen	5	12.10%

Table 1.1 Year of Study for the Entire Class

Table 1.2 shows the overall demographic results for the students' discipline/major. The data shows that the majority of students are pursuing a Bachelor of Science (BS) in Business Administration (BSBA) degree with a specialty focus. Only 6% of the students enrolled in the applications/programming course are enrolled in Bachelor of Arts (BA) a degree program and 9% are enrolled in a degree program outside the college of business. This course enrollment data is not unusual for this applications/programming course because this course is taught by an ITM faculty within the college of business. This course is also an approved elective for college of business students.

	Frequency	Percent
BSBA Management	5	12.10%
BSBA ITM	13	39.40%
BSBA Accounting	2	6.10%
BSBA Marketing	2	6.10%
BA History	1	3.00%
BS Digital Forensics	1	3.00%
BSBA Finance	1	3.00%
BA Communication Studies	1	3.00%

Table 1.2 Major of Study for the Entire Class

Descriptive Statistics for the Collected Data Set

While the above demographic data describes the students enrolled in this course, it is important to note that the only demographic data collected from the survey was gender and age. Moreover, out of the overall sample size of 33 students, only 28 students completed the survey.

Table 2.1 reports the gender and age of the students that completed the survey. The majority of students (n=21) completing the survey were male. The majority of students (n=27) were traditional aged students. Only one student was non-traditional aged. This study defined traditional students as 18 to 24 years of age and non-traditional students as 25 plus years of age Greater than 24 years of age.

	Frequency	Percent
Male	21	75.00%
Female	7	25.00%
18-24	27	96.40%
>24	1	3.60%

Table 2.1 Gender and Age of the Survey Participants

Descriptive Statistics and t-Test Results of the Data Set

Descriptive Statistics were used to answer the research question 1 and 2. Table 3.1, located in the appendix, provides detailed questions responses. Specifically, the majority of students' responses are within the strongly agree and agree categories. These results suggest that the majority of students had a positive perception and experience with using group work in developing a mobile application. Additionally, Table 3.2, also located in the appendix, provides the mean scores for students' perceptions regarding hands-on-on group when developing a mobile application. The mean scores were all above 3.75 with the majority of mean scores above 4.0 "Agree". However, the perceived learning for question 8 was the weakest with respect to agreement.

Table 3.2 also shows the mean score and standard deviation for the dataset grouped into the four categories. The four categories are a measurement of effectiveness for confidence in quality, perceived productivity, enjoyment, and perceived learning. Confidence in quality was the mean score for the grouping of questions/statements 1, 2 and 3. Perceived productivity was the mean score for the question/statement 4. Enjoyment was the mean score for the grouping of question/statements 5, 6 and 7. Perceived Learning was the mean score for the grouping of question/statements 8, 9 and 10.

The results for each category also has mean scores close to or above 4.0 (Agree). This indicates that students agree that they are producing quality, are productive within their group, and enjoy group work when developing a mobile application.

While the mean scores and standard deviations provide insight into the students' perceptions, an effective measurement of the category outcomes is to test for a significant difference between the each of the four category outcomes. To answer research question 3, a paired t-test was performed on the data set. The results of the paired t-test indicated that there were no significant differences among any of the category outcomes. One can conclude that there is no significant difference because the four category outcomes are very close in score.

Additional statistical analyses were conducted to answer research question 4. To test the significant difference between gender and the four category outcomes, an independent t-test was performed. Results indicated that there was no significant difference between gender and confidence in quality, enjoyment or the perceived learning categories. However, there was a significant difference between the perceived learning category outcome's mean scores for males ($M=3.87$, $SD=1.09$) and female students ($M=4.52$, $SD=.42$), $t(26)=1.52$, $p=.032$). Specifically, females had a greater perceived learning in using group work to develop a mobile application in Stencly. Please reference Tables 4.1 and 4.2 in the appendix for details.

5. CONCLUSION AND LIMITATIONS

The results indicated that students have positive perceptions regarding using group programming for mobile application development. Our results are consistent with the results of similar studies that utilized a collaborative learning technique or a pair learning techniques. However, this research is important because as programming classes continue to become more interdisciplinary, the more important it is for educator's to engage and challenge all levels students using savvy mobile application software to further apply key programming concepts learned. Additionally, by having the student work in groups the instructor is making the students responsible for having a successful learning experience.

This study is not without limitations. This study had a small sample size and made no attempt to control for variables that may impact student perception of group work for mobile application development. Additionally, this study did not analyze if group size affected the students' responses. Therefore, it is uncertain if group size mattered. Additionally, students were surveyed after they presented their group's mobile application to the entire class. Prior to taking the survey, students received feedback from their instructor and classmates. Therefore, it is uncertain if the students honestly answered the questions or answered the questions based upon the instructor and classmate feedback. Furthermore, because the survey was anonymous, there was no way to test the differences between ITM and non-ITM students or working group size.

Nevertheless, this study demonstrated group programming for mobile application development can be used as a method to increase learning outcomes of a hands-on programming course. Future research should better control variables for construct validity. Additional research should be conducted with a larger sample size from various hands-on courses with several mobile application development tools in various computer lab environments over an extended period of time. Finally, future research should also be conducted on the effect of group size, as well as whether or not students who prefer group work actually do better when given that option versus students who are force to do group work against their preference.

6. REFERENCES

Babb, J. S., Longenecker, B., Baugh, J. & Feinstein, D. (2014). Confronting the issues of programming in information systems curricula: The goal is success. *Information Systems Education Journal, 12*(1), 42-72.

Baer, J., 2003. Grouping and achievement in cooperative learning. *College Teaching*, *51*(4), 169-174.

Burd, B., Barros, J. P., Johnson, C., Kurkovsky, S., Rosenbloom, A., & Tillman, N. (2012). Educating for mobile computing: Addressing the new challenges. *In Proceedings of the Final Reports on Innovation and Technology in Computer Science Education* 2012 Working Groups *Conference* (ITiCSE-WGR '12), New York, NY, USA, 51-63.

Chigona, W., & Pollock, M. (2008). Pair programming for information systems students new to programming: Students' experiences and teachers' challenges. *PICMET 2008 Proceedings,* 1587-1594.

Ching-Chiu, C. (2006). An investigation of learning style differences and attitudes toward digital game-based learning among mobile users. *In Proceedings of the Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education (WMTE '06)*, Washington, DC, USA, 29–31.

Chinn, C. & Chinn, L. (2009). Collaborative learning. Retrieved July 1, 2015 from http://www.education.com/reference/article/collaborative-learning/

Cliburn, D. C. (2003). Experiences with pair programming at a small college. *Journal of Computing Sciences in Colleges, 19*(1), 20-29.

Ebrahimi, A. (1994). Novice programmer errors: Language constructs and plan composition. *International Journal of Human Computer Studies, 41*(4), 457-480.

Faja, S. (2014). Evaluating effectiveness of pair programming as a teaching tool in programming courses. *Information Systems Education Journal, 12*(6), 36-45.

Hall, B. (2013). Teenagers & smartphones: How they're already changing the world. Retrieved July 1, 2015 from: http://pewinternet.org/Media-Mentions/2013/Teenagers-Smartphones-How-Theyre-Already-Changing-The-World.aspx.

Hoffman, B. (2014). Interdisciplinary group projects in game design and mobile application development. *Journal of Computing Sciences in Colleges, 26*(6), 105-106.

Howard, E. V. (2007). Attitudes on using pair-programming. *Journal of Educational Technology Systems, 35*(1), 89-103.

Hu, H. H. & Shepherd, T. D. (2014). Teaching CS1 with POGIL activities and roles. In Proceedings of Computer Science Education Conference (SIGCSE'14), Atlanta, GA, USA. 127-132.

Jenkins, T. (2002). On the difficulty of learning to program. *In the Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'01), 54-56.*

Kinnunen, P., Mccartney, R., Murphy, L. & Thomas, L. (2007). Through the eyes of instructors: A phenomenographic investigation of student success. In *Proceedings of the 3rd International Workshop on Computing Education Research Conference,* 61-72.

Klopfer, E. (2008) *Augmented Learning: Research and Design of Mobile Educational Games.* MIT Press, Cambridge, MA

Lasserre, P. (2009). Adaptation of team-based learning on a first term programming class. *In Proceedings of the 14th Annual ACM SIGCSE Conference on innovation and Technology in Computer Science Education* (Paris, France, July 06 - 09, 2009). *ITiCSE '09*, New York, NY, 186-190.

Lasserre, P. & Sztostack, C. (2011). Effects of team-based learning on a CS1 course. *In Proceedings of the 16th Annual ACM SIGCSE Conference on innovation and Technology in Computer Science Education* (Darmstadt, Germany, July 27-29, 2011). *ITiCSE '11*, Darmstadt, Germany, 133-137.

Nunnaly, J. (1978). *Psychometric Theory*. New York: McGraw-Hill.

Madden, M., Lenhart, A., Duggan, M., Cortesi, S., & Gasser, U. (2013). Teens and technology 2013. Retrieved July 1, 2015 from: http://pewinternet.org/Reports/2013/Teens-and-Tech.aspx.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communication of the ACM, 49(8),* 90-95.

McKeown, J. (2004). Team game programming assignment: Nifty assignments. *In the Proceeding of the Consortium for Computing Sciences in Colleges.* 191

Medina, R.D., Gomez-Perez, D., Neito-Reyes, A. & Santos, C. B. (2013). A method to form learners groups in computer-supported collaborative learning systems. *In Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality,* Salamanca, Spain, 261-266.

Mentz, E., Van der Walt, J., & Goosen, L. (2008). The effect of incorporating learning principles in pair programming for student teachers. *Computer Science Education, 18*(4), 247-260.

Michaelsen, L. K., Knight, A. B., & Fink, D. L. (2004). *Team-based learning: A Transformative Use of Small Groups in College Teaching*. Stylus Publishing. Sterling VA.

Powell, L. M. & Wimmer, H. (2015). Evaluating the effectiveness of self-created student screencasts as a tool to increase student learning outcomes in a hands-on computer programming course. *Information Systems Education Journal, 13*(5), 106-111.

Santos, R. A. (1999). Cronbach's Apha: A tool for assessing the reliability of scales. *Journal of Extension, 37*(2).

Sleeman, D. (1986). The challenges of teaching computer programming. *Communications of ACM, 9*(29), 840-841.

Smith, A. (2013). Smartphone ownership – 2013 update. Retrieved July 1, 2015 from: http://www.pewinternet.org/~/media//Files/Reports/2013/PIP_Smartphone_adoption_2013.pdf.

Williams, L. & Kessler, R. (2001). Experiments with industry's "pair-programming" model in the computer science classroom. *Computer Science Education, 11(1),* 7-20.

Zacharis, N. Z. (2011). Measuring the effects of virtual pair programming in an introductory programming java course. *IEEE Transactions on Education, 54*(1), 168-170.

APPENDIX

Table 3.1 Percentage for Student Responses

Outcomes and Questions	Strongly Agree (5)	Agree (4)	Neither Agree nor Disagree (3)	Disagree (2)	Strongly Disagree (1)
Confidence in Quality					
1. I find that group programming develops better mobile application than developing myself.	25.0%	46.4%	5.0%	0.0%	3.0%
2. More errors were found and fixed with group programming.	35.7%	50.0%	3.6%	3.6%	7.1%
3. I was more confident in the work with group programming.	42.9%	46.4%	3.6%	0.0%	7.1%
Perceived Productivity					
4. The mobile application was developed quicker because of group programming.	32.1%	42.9%	10.7%	10.7%	3.6%
Enjoyment					
5. I enjoy programming / developing mobile applications with a group more than programming/developing alone.	35.7%	46.4%	7.1%	7.1%	3.6%
6. If I had a choice, I would work in a group again.	42.9%	42.9%	7.1%	3.6%	3.6%
7. I liked using group programming during the in-class labs.	39.3%	50.0%	7.1%	0.0%	3.4%
Perceived Learning					
8. I have learned more from doing the work because of group programming.	28.6%	39.3%	17.9%	10.7%	3.6%
9. It was helpful to discuss programming problems and solutions with my group.	50.0%	35.7%	3.6%	7.1%	3.6%
10. I think that using group programming during the in-class labs helped me better understand the concepts.	42.9%	35.7%	14.3%	3.6%	3.6%

Table 3.2 Question Mean and Standard Deviation

Outcomes	Questions	Mean	Standard Deviation
Confidence in Quality		3.99	0.97
	1. I find that group programming develops better mobile application than developing myself.	3.75	1.18
	2. More errors were found and fixed with group programming.	4.04	1.11
	3. I was more confident in the work with group programming.	4.18	1.06
Perceived Productivity		3.89	1.10
	4. The mobile application was developed quicker because of group programming.	3.89	1.10
Enjoyment		4.14	0.92
	5. I enjoy programming/developing mobile applications with a group more than programming/developing alone.	4.04	1.04
	6. If I had a choice, I would work in a group again.	4.18	0.98
	7. I liked using group programming during the in-class labs	4.21	0.88
Perceived Learning		4.04	1.00
	8. I have learned more from doing the work because of group programming.	3.79	1.10
	9. It was helpful to discuss programming problems and solutions with my group.	4.25	1.01
	10. I think that using group programming during the in-class labs helped me better understand the concepts.	4.11	1.03

Table 4.1 Category Means and Standard Deviations by Gender

Category	Mean	Standard Deviation
Confidence in Quality		
Male	3.95	1.08
Female	4.10	0.57
Preceived Productivity		
Male	3.81	1.21
Female	4.14	0.69
Enjoyment		
Male	4.03	1.10
Female	4.48	0.47
Preceived Learning		
Male	3.87	1.09
Female	4.52	0.42

Table 4.2 Results of T-test

Category	Sig.	T	Df	Sig (2-tailed)	Mean Difference
Confidence in Quality					
Equal variance assumed	.511	-.333	26.00	.742	-.143
Equal variance not assumed		-.449	20.25	.658	-.143
Perceived Productivity					
Equal variance assumed	.141	-.687	26.00	.498	-.333
Equal variance not assumed		-.898	18.69	.380	-.333
Enjoyment					
Equal variance assumed	.290	-1.11	26.00	.275	-.444
Equal variance not assumed		-1.58	22.78	.129	-.444
Perceived Learning					
Equal variance assumed	.078	-1.52	26.00	.140	-.651
Equal variance not assumed		-2.27	25.08	.032	-.651