Work in Progress - Unexpected Student Outcome from Collaborative Agile Software Development Practices and Paired Programming in a Software Engineering Course

Chih-wei Ho[1], Kelli Slaten[2], Laurie Williams[3], and Sarah Berenson [4]

Abstract - **There has been low representation of women in Computer Science. This paper describes the initial findings of a three-year research project about women in the field of information technology. The goal of this research is to examine the effect of pair programming and agile software development on students. During the first semester of this project, pair programming was used in a junior/senior software engineering class at North Carolina State University. In this paper, we share the grounded theory analysis of three interviews and thirteen project retrospective essays of the female students. Theoretical models were developed to describe (a) the factors of students' enjoyment in a software design course that employs agile software methods, (b) context that influenced students' study habits, and (c) the effectiveness of pair programming and agile methods. Initial findings indicate that pair programming is an effective practice for the female students, but it also brings new challenges for the instructors.**

Index Terms - Pair programming, software engineering education.

INTRODUCTION

In pair programming, two programmers work at one computer on the same programming task. One of the programmers is the driver, who controls the keyboard and mouse, and actively performs the programming task. The other, called the navigator, watches the driver's work and acts as a brainstorming partner. Pair programming shows several promising properties [3] for educational purpose. In the pilot phase of a three-year research project, a qualitative study was conducted to develop a theory about the effect of pair programming on female computer science students.

RESEARCH METHODS

To explore the variables regarding the use of agile software practices in a classroom, we use a qualitative approach called grounded theory to guide the data collection and analysis process. Grounded theory, first described by Glaser and Strauss [1], has a three-step coding process. Open coding is used to label the data with different categories; axial coding is used to identify the interrelationships among the categories; and selective coding is used to generate theories from the categories of interest.

We used grounded theory to analyze interviews and students' retrospective essays. We tailored the coding process for textual data. Following is a brief description of the process:

- **Open Coding:** The goal of this step is to find out the categories and the properties of each category. The textual information is transformed into pairs of <Category, Excerpt>. This step also helps to reduce the data for further analysis.
- **Axial Coding:** We use cognitive maps [2] to depict the interrelationships among the categories found in open coding. Figure 1 is an example of cognitive map, in which a box represents a category, solid arrows are positive influences, and dashed arrows are negative influences.

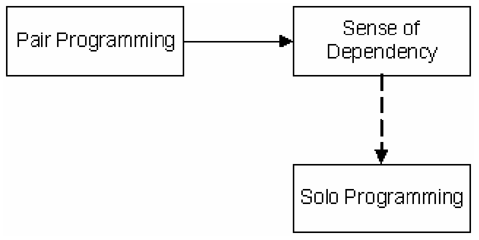

FIGURE 1
AN EXAMPLE OF COGNITIVE MAP

- **Selective Coding:** In the last step, we focus on the categories of interest. We observe the patterns that are related to these categories, and form a theory from the pattern. Sometimes we need to form the union of the patterns of several cases to have a broader view of the same phenomena.

INITIAL FINDINGS

[1] Chih-wei Ho, North Carolina State University, Department of Computer Science, cho@unity.ncsu.edu
[2] Kelli Slaten, North Carolina State University, Department of Math and Science Education, kmslaten@ncsu.edu
[3] Laurie Williams, North Carolina State University, Department of Computer Science, williams@csc.ncsu.edu
[4] Sarah Berenson, North Carolina State University, Department of Math and Science Education, berenson@unity.ncsu.edu

In the first semester of the project, pair programming was used in a junior/senior software engineering class at North Carolina State University. Fifteen female and 76 male students in that class participated in this research. There were three programming assignments. The first two were paired assignment, in which a student was assigned a partner. The third was a solo assignment, in which each student worked on his or her own. After the assignments, there was a six-week group project to develop a software reliability estimation plug-in for an open-source Java development environment. In the project, there were four to five students in each group. There were two types of groups for the project: paired and solo. In a solo group, each team member was assigned a piece of the overall project, programmed alone and integrated completed work. In paired groups, the members practiced pair programming, where the team was divided into pairs and each programming task was assigned pair-wise. At the end of the project, the students were asked to write a retrospective essay. We interviewed eight students, three of which were female, at the beginning of term project. We analyzed the retrospective essays and interviews of the female students. Below are the initial results:

I. Project Enjoyment

In the retrospectives, about 50% of the female students said they enjoyed the project, and the other 50% said they did not. Those who did not enjoy the project said that the technology used in this project was new to them, and it was not covered by the course material. They did not like it because they had to spend extra time learning it on their own. Yet some other students thought that self-learning was a pleasure itself.

Some students said the usefulness of the project also brought the enjoyment. It is interesting because some students also thought that this project was not useful at all, and thus did not like the project. Therefore, when a project assignment is given to the students, instructors can spend some time telling the students why the assignment is important. The students may just have different attitudes when doing the project.

Another factor that affected project enjoyment was teamwork. The students liked a good team leader and balanced workload. This can be achieved by giving the students enough collaborative skills.

II. Study Habits

The female students showed strong caring for other people. Caring for people brought them some pressure when doing pair programming, because they did not want to let their partners down. We call this effect *pair pressure*. Pair pressure had two different effects on female students. It gave them the sense of responsibility. When they made their pairing schedule, they responsibly adhered to it because other people counting on them. This helped them start the work earlier and improved their time management skills. However, it also made the students dependent on their partners. In the third assignment, which was a solo one, this pair pressure was gone. As a result, some students said they did not start the assignment until very late.

III. Pair Programming Effectiveness

When programming in pairs, the navigator constantly reviews the driver's work. This is called *pair review*. The students said having another person watching over their shoulders helps them debug their program. Discussing the work with the partner also made problem solving easier.

When working with a partner, the females did something they indicated they would not do if working alone. We call this effect *pair courage*. The students said that when they had problems, it was easier for them to ask the lab assistants if they had companions because they knew they were not the only one who had a question or did not understand something.

There were two things that made pair programming less effective for the female students. First, the students all had different schedules. It was difficult for them to find some common time to do pair programming. The result was that the students in pair groups spent no more than 70% of their time programming in pairs. Second, bad pairing experiences caused some students to form negative feeling about pair programming. Some female students said, when pairing with male students, their partners did not listen to them. Additionally, sometimes their partners did the entire job or did not do anything at all.

FUTURE WORK

The initial findings show the effects of pair programming on female students and also bring up some difficulties to use pair programming in a classroom. We find out the following problems that need to be addressed in future study:

- How do we improve the students' collaborative skills?
- How do we avoid the negative effect of pair programming, i.e. sense of dependency?
- How do we make sure the workload equity in group assignments?

The study is expanding to include North Carolina A&T and Meredith College. We are conducting more qualitative and quantitative inquiries to make the findings more reliable. From the qualitative perspective, more observations and interviews will provide more supportive or contrary evidence and a richer context of the phenomena. Quantitative experiments are also needed to test the generality of the findings and to provide triangulated results.

REFERENCES

[1] Glaser, B. G. and Strauss, A. L., *The Discovery of Grounded Theory: Strategies for Qualitative Research*, 1967.

[2] Ryan, G. W. and Bernard H. R., "Data Management and Analysis", in *Handbook of Qualitative Research 2nd Edition*, 2000.

[3] Williams, L. and Kessler, R. R., *Pair Programming Illuminated*, 2002.