Improving PSP Education by Pairing: An Empirical Study

Guoping Rong  
State Key Laboratory for Novel Software Technology  
Software Institute  
Nanjing University  
Nanjing, P.R.China  
ronggp@software.nju.edu.cn

He Zhang  
National ICT Australia  
University of New South Wales  
NSW, Australia  
he.zhang@nicta.com.au

Mingjuan Xie, Dong Shao  
Tao Zheng  
Software Institute  
Nanjing University  
Nanjing, P.R.China  
{mg1132013, dongshao,zt} @software.nju.edu.cn

Abstract—Handling large-sized classes and maintaining students’ involvement are two of the major challenges in Personal Software Process (PSP) education in universities. In order to tackle these two challenges, we adapted and incorporated some typical practices of Pair Programming (PP) into the PSP class at summer school in Software Institute of Nanjing University in 2010, and received positive results, such as higher students’ involvement and conformity of process discipline, as well as (half) workload reduction in evaluating assignments. However, the experiment did not confirm the improved performance of the paired students as expected. Based on the experience and feedbacks, we improved this approach in our PSP course in 2011. Accordingly, by analyzing the previous experiment results, we redesigned the experiment with a number of improvements, such as lab environment, evaluation methods and student selection, to further investigate the effects of this approach in PSP education, in particular students’ performance. We also introduced several new metrics to enable the comparison analysis of the data collected from both paired students and solo students. The new experiment confirms the value of pairing practices in PSP education. The results show that in PSP class, compared to solo students, paired students can achieve better performance in terms of program quality and exam scores.

Keywords—personal software process; collaborative learning;

I. INTRODUCTION

Personal Software Process (PSP), designed by Watts S. Humphrey (from CMU/SEI), is a framework of techniques to help engineers improve their performance [1]. As Humphrey stated, software engineers can enjoy benefits, such as fewer code defects, better estimation and planning, enhanced productivity by learning and using the disciplines of the PSP [1]. What’s more, measurement and analysis tools included in PSP framework can also help software engineers understand their own capabilities and improve personal performance. Will Hayes et al. conducted an empirical study of the impacts of PSP on individual software engineers during PSP training, the results indicated that improvements can be expected in four dimensions: size estimation accuracy, effort estimation accuracy, product quality and process quality were statistically significant [2].

Due to the advantages mentioned above, it is believed that delivering PSP course to students in universities can help to train qualified software engineers who meet industry’s requirements better. As a result, most software schools in China provide similar courses. However, delivering PSP course in China often faces a big challenges resulted by the rapidly increasing demand for skilled software engineers due to the fast growth of IT industry. Based on an experiment conducted at Nanjing University in 2010 [3], two major challenges for PSP course in education environment were discovered: the size of class and students’ participation and motivation, and it is also suggested in [3] that introducing several practices of Pair Programming (PP) to PSP course may tackle these challenges. The results of our experiment conducted in 2010 (EXP2010) indicated that the performance of paired students in assignments are significantly better than those solo students. However, the program (work product) quality (measured by number of defects) and the effectiveness of education (by scores in final exam) have no significant difference between the two groups. This motivated us to re-investigate the whole process to deliver PSP course with a more rigorously designed experiment.

Accordingly, we not only further improved the proposed PSP education approach, such as learning environment and evaluation methods, in the course in 2011 summer school, but also improved the design of EXP2010 and conducted a new experiment (EXP2011) to further investigate impact of this new PSP education approach on students’ overall performance. After data collection and analysis, the results show statistical significance that the paired students performed better than the solo students in both assignments and exam. In addition, we also report our experiences from EXP2011 and suggestions for improving PSP training in academic education environments.

II. BACKGROUND AND RELATED WORK

A. Personal Software Process in Education

Since its introduction, PSP has been used in university education to provide students with self-improvement software process. It can serve as a part of other courses, or as a standalone course. In one experiment [4], PSP is used throughout the whole curricula. PSP can be contained in a series of courses related to software engineering. There also exist some experiments that integrated some typical practices of PSP into traditional programming courses [5, 6, 7]. In all these cases, the PSP practices need to be tailored to meet various objectives of different courses as well as time constraints [4, 6]. Results from these attempts show that PSP brought some positive impacts to these courses, e.g., it can
help solve the problems of poor estimation and planning [4, 7], reduce debugging time and improve quality [6], and increase productivity [9]. However, incomplete delivery of PSP course will inevitably impact the training effect to meet objectives of PSP course. For instance, the attempts in education mentioned above only provide training on basic PSP process elements such as time log and defect log, which contribute limited help for quality management at individual level. Besides, while delivering the PSP course in combination with other courses, students may pay more attention on learning algorithms and languages than PSP discipline [5]. In an example, PSP was optional for students when practicing programming language, which resulted in a considerable number of students quitting PSP in the end [11]. When PSP was made as a standalone course, students have to master at least one programming language before enrolling PSP course. Results of the PSP course vary in different studies, some of which reported positive effects [1, 9, 10], such as improved effort estimation skills and product quality. However, in one case [8], students in PSP course failed to achieve significant improvements, such as size and effort estimation, and only two third (20 out of 31) students passed the course. While in all these cases with full PSP courses delivered, their class size remains small.

B. Pair Programming in Education

Pair Programming (PP) has been also widely adopted as a pedagogical tool in higher education. With a series of reported experiments, researchers found that PP has many positive effects on software development. On the social side, it helps form more active laboratory atmosphere than the traditional laboratory; under such friendly atmosphere, students will be more productive, have less frustration [12,13] and enhance self-confidence and enjoyment [13, 14, 15]. On the technical side, PP can help improve the individual's coding ability [14, 16], improve design quality [13], reduce the number of defects [14, 15], reduce risks [14], and enhance the quality of the final program [13] by establishing collaboration among students. On the educational side, PP can also reduce number of cheating cases [13], help reduce the workload of educators [13], and ultimately improve the quality of teaching.

However, there are many factors that may affect the final teaching effects when PP is applied. For example, the working environment may affect the results of PP. Williams [18] suggested that the pairs should be able to comfortably sit next to each other as they work. Williams et al. [19] reported that 96% programmers agreed that appropriate workspace layout was critical to their success. Wallace et al. [20] also believed that the sitting arrangement was critical to the success of PP. Meanwhile, proper feedback mechanism such as assignment evaluation can also impact the effectiveness of PP. Therefore, as Williams suggested, the teaching staff should give guidance to encourage pairs to find answers on their own, rather than providing them with answers [18].

C. PSP Course in Nanjing University

Software Institute of Nanjing University is one of 37 national pilot software schools established in China in 2002. It receives around 300 new freshman students annually. In 2006, PSP course became an elective course for undergraduate students. To ensure the course quality, it is delivered by one SEI-certified PSP Instructor and follows SEI’s standard. Since 2010, the course has turned to be a compulsory course due to the recognition of the importance of process related skills for software engineers. It means the size of an annual PSP class increased up to 300, which became a big challenge to the PSP course, particularly in assignments evaluation. With reference to the reported experiences and empirical studies, in 2010 summer school, we conducted an initial experiment by introducing some PP practices, e.g., forming pairs for assignments, into PSP course and arranged some students finish the PSP assignments in pairs. We compared the assignment results from both paired students and solo students, which indicated that learning PSP in pairs can reduce the amount of homework without sacrificing the effect and quality of PSP education [3].

However, the feedbacks from both groups indicate some outstanding issues in EXP2010, including mutual interference among pairs, difficulty in understanding evaluation remark, and the inability of tracking corrections over submissions. For solo students in PSP course, they were asked to finish the assignments without communication to each other in lab. However, for paired students, who need to communicate to each other in all the phases, everyone might be distracted in a noisy environment. Students interviewed after EXP2010 reflected that such an environment had been seriously affected their practicing PSP.

In addition, the teaching assistant (TA) groups were designed to switch in evaluating students’ submissions in EXP2010, so that the students had each assignment evaluated by different TA groups. This arrangement, however, disabled TAs to track correction over students’ assignment submissions, and lowered their evaluation consistency. For example, TA group may not care much about whether students corrected the mistakes in the prior four assignments. It led inconsistent understanding of PSP process for students. Some students mentioned that they constantly misunderstood the remarks from different TA groups. These findings to a certain extent explained the reason why no significant score difference between paired students and solo students was observed in the final exam of EXP2010.

In 2011, we made improvements on the reflected issues in PSP course, and redesigned the experiment to revisit the effects of learning PSP in pairs, and conducted the new experiment (EXP2011) in the summer school. Section III elaborates the (paring) education approach with improvements based on EXP2010. Section IV describes the design and arrangement of EXP2011, especially the improvements from EXP2010. Section V presents analysis results and findings from EXP2011. Section VI discusses the related issues of the study and gives some suggestions for delivering full PSP course. It is followed by the threats to
validity of this study in Section VII. The paper is concluded in Section VIII with suggestions for future research.

III. PSP EDUCATION USING PAIRING PRACTICES

A. Original Course Design

1) Timing

In Nanjing University, the PSP course started in the summer school after the first academic year ended. The considerations are two-fold; (1) students may already possess preliminary programming language skills for the PSP course at the end of their first academic year, so they do not need to spend much time on learning language during PSP assignments; (2) the required 10 continuous days for a full PSP education is only available in the summer school, during which all the students enrolled in the course in full time.

2) Roles

Students enroll in the PSP course when they finish their first academic year in university. The number of enrolled students is typically between 260 and 280 each year in Nanjing University.

PSP instructor’s major responsibilities in the course are two-fold: to deliver lectures and to lead evaluation.

Teaching assistants (TAs) are selected from sophomore students who performed well in PSP course in last year. Their major responsibility is to evaluate the students’ submissions. They need to find deviations to PSP process discipline in the students’ submissions and check that students had corrected all the issues. Therefore, each submission may be evaluated in several rounds by the teaching assistant.

3) Lecture and Practice

| TABLE I. TYPICAL DAY FOR STUDENTS IN PSP CLASS |
|-----------------|------------------|------------------|
| Session | Time | Content |
| Lecture | 09:00 ~ 12:00 | Class lecture |
| Lab | 13:30 ~ 22:00 (3 batches) | Lab practice, assignment |

| TABLE II. PSP COURSE CONTENT |
|-----------------|-----------------|------------------|
| Day # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Lecture # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Program exercise # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Schedule and content for PSP education are similar to SEI’s standard course arrangement. We use classroom lectures combining laboratory practices to deliver the PSP course in summer school. PSP instructor gives all the lectures and explains the content of each lecture to the whole class in one big theater with capacity of more than 300 students in the morning sessions. After that, in the afternoon-evening sessions students are given the requirements of each assignment and work on their assignments in three large labs by following a pre-defined PSP process according to the requirements. Each student has his/her assigned seat and PC in the laboratories. All the assignments must be completed and submitted before the end of the day. Table I shows the schedule of one typical day in the course.

We use SEI standard ten-day PSP course material [22] to deliver the course. The arrangements for the entire course are shown in Tables II. The serial numbers used in the table is same as that in SEI’s course materials. And there are no program exercises on Day 5 and 10.

| TABLE III. DIFFERENCE TO TRADITIONAL PP |
|-----------------|-----------------|------------------|
| Objective | to develop software products. | Pairing in EXP2011 |
| Roles Swap | happen whenever needed. | happen only between assignments. |
| Task for observer | to provide different views and thoughts for driver, usually focus on technical aspects, (e.g., new design and implementation strategy). | to provide process assurance for driver, usually focus on managerial aspects and process disciplines |

Pairing practices need special arrangement and guidance in PSP education. According to the positive results from EXP2010, most students in the course were required to complete all assignments in pairs. For paired students, each had a fixed partner during the entire course. However, pairing only happened in the lab session when students were required to finished exercises. For instance, in one pair (Student A and B), if the exercises in odd numbers (#1, 3, 5 and 7) were completed by A (the ‘driver’), B acted as the ‘observer’. Then in even numbers (#2, 4, 6 and 8), A and B swapped their roles. The swap must be validated by teaching assistants, so as to avoid the case that all the assignments are completed by only one student.

Table III listed the major differences between traditional PP and the pairing practices in our PSP education approach. We provided the paired students brief guidelines to guide their practice in lab. For a single pair (Student A, B), we suppose that Student A acts as the driver and B as the observer, i.e. A turns to complete the assignment. The work in this case is exactly the same as that A completes the work individually, while B’s main task is to participate in the discussion and verify whether A carries out the work in accordance with the process standards, e.g., recording time log and defect log, in all PSP phases, i.e. planning, detailed design, coding, unit testing and postmortem. The detailed tasks of the roles in pairs in each specific phase are...
elaborated in table IV. However, only the driver was required to collect process data.

4) Assignment and Evaluation

Assignments are required to be completed after the daily lecture. As presented in table II, except for the reports, students were required to write a program by following a predefined level of PSP process. Usually, students should finish and submit their assignments by the end of the day when they receive them.

Prior-evaluation training: Assignment evaluation and feedback are extremely important for students to master the PSP discipline. Hence the quality of the evaluation plays a vital role in PSP course. The training before evaluation will help the TA team fully understand the assignment requirements and evaluation criteria. To guarantee the same standard and quality of the evaluation, we used the evaluation checklist from SEI’s course material. At the beginning of this training, the PSP instructor made an introduction and demonstration about the use of the checklist to the whole evaluation team. In this stage, the whole team was separated into several small groups with 2 TAs in each. Then the PSP instructor randomly selected several students’ submissions as samples and led the whole team to evaluate the first submission, and then the TA groups began to evaluate the rest samples, while the PSP instructor made necessary corrections. This approach may help the evaluation team quickly achieve a consistent understanding of the checklist.

Post-evaluation meeting: After each evaluation, the evaluation team also met for communicating the results and sharing their experiences. The typical topics at the meeting include (but not limited to):
- What are the typical issues in the students’ submissions?
- Are there any cases in which the evaluation group feels difficult to reach consensus?
- Are there any special circumstances which are worthy discussing?

Through such discussions, the evaluation team could further uniform their evaluation criteria.

B. New Improvements

Focusing on the issues identified after PSP course in 2010, we made improvements to this education approach in 2011 from two aspects, i.e. lab environment and evaluation mechanism, which are described below respectively.

1) Lab Environment: First, students were separated into several batches to finish the assignments, which allows enough space to ensure at least 2 meters distance between groups using the same labs as in 2010 PSP course. Also the students were required to keep discussion softly. With this arrangement, the mutual interference between two groups was minimized. The students’ feedbacks after the course also confirmed that they were not interfered by others. Second, in order to keep students’ focus on the assignments, we provided standardized hardware and software environment with the same configuration to every student. In lab sessions, Java and Eclipse were used as the only programming language and IDE. Office 2003 was used for reporting and documenting. This standardized lab environment shortened students’ lab time from 180 to 150 minutes, and provided more flexibility to arrange lab session for students than in 2010. For example, it will be easier to separate all the students into three batches to do assignments from 1:30 pm to 10:00 pm. Hence, the space (density) needed between paired students could be well guaranteed. Besides, with standardized facilities, the logistic effects to the recorded process data (e.g., time spent in configuring Eclipse, defects caused by wrong version of JRE) could be well controlled.

2) Evaluation Mechanisms: To ensure each assignment evaluation was complete and consistent, we still used the standard evaluation checklist provided by SEI. However, compared with the 2010 course, we have made two important adjustments in 2011. First, we asked the assessors (TAs and Instructor) must give instructive remarks during evaluations. Differring from the evaluation in 2010 which only indicates the mistakes in submissions, the instructive evaluation needs to address: 1) what’s wrong in the submission? 2) what correct situation should be? and 3) the reason. This information provided students the guidance to understand and master the concepts and practice of PSP. For example, when a novice student first applies the PROBE estimation method, he/she is easy to mistake the relation among the different types of code size, e.g., actual size of new code (A) and modified code (M) on different forms. In the 2010 course, TA just pointed out the mistakes, which may provide very limited help for students’ understanding. With instructive evaluation, however, TA should also explain the reason for why a certain relation exists, e.g., (A+M) in the estimating form only includes those code within methods in a Java class, while (A+M) in the project plan summary form also includes those code outside methods. This evaluation method can be more effective in helping students understand PROBE estimation method. This was confirmed by the students’ feedbacks after the course that the instructive evaluation is very helpful to understand and master the PSP principles and practices. Second, in order to ensure the evaluation consistence on PSP process discipline, we cancelled the rotation between TA groups. A fixed TA group was assigned to evaluate each student or student pair. The resulting consistent remarks from the same TA group improved students’ understanding. This approach also allows the TA group tracking the change history of each assignment to ensure that all discovered issues had been understood and resolved.

IV. RESEARCH OBJECTIVES AND DESIGN

A. Research Objectives

The primary research objective of the new experiment is to investigate the effectiveness and efficiency of PSP training
by pairing students in tertiary education. We also expect to identify improvement opportunities for future PSP education through the experiment. The Goal/Question/Metric (GQM) [21] approach provides a systematic way to detail the research objectives and define corresponding metrics. Table V presents the GQM definition for this experiment.

<table>
<thead>
<tr>
<th>Goal(s)</th>
<th>Question(s)</th>
<th>Metric(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>To investigate the effectiveness of PSP education by pairing</td>
<td>Do paired students following process discipline better than solo students when applying PSP?</td>
<td>Measure the number of inconsistent items in the predefined submission checklist when evaluating assignment submissions from paired students and solo students.</td>
</tr>
<tr>
<td></td>
<td>Do paired students produce programs with better quality than solo students?</td>
<td>Measure the number of residual defects in both paired students’ and solo students’ programs.</td>
</tr>
<tr>
<td></td>
<td>Do paired students achieve higher scores than solo students in the final exam?</td>
<td>Measure the scores in the final exam of both paired students and solo students.</td>
</tr>
<tr>
<td>To investigate the efficiency of PSP education by pairing</td>
<td>Do paired students require more time than solo students in PSP education?</td>
<td>Measure the time used to finish all the assignments for both paired students and solo students.</td>
</tr>
<tr>
<td>To identify improvement opportunities for future PSP education</td>
<td>What are the most necessary improvements in PSP education?</td>
<td>Measure the category of inconsistent items and number of inconsistent items fall into each category.</td>
</tr>
</tbody>
</table>

Based on evidences discovered in EXP2010, our approach that incorporates typical practices of PP into PSP education is able to improve the effectiveness and efficiency of the education process. According to Table V, we set up four propositions for EXP2011:

P1: Paired students can follow PSP process discipline better than solo students in assignments with fewer inconsistent items in their submissions.

P2: The submitted programs by paired students have fewer defects than solo students.

P3: Paired students can achieve higher scores in exam than solo students in PSP course.

P4: The needed assignment time has no significant difference between paired students and solo students.

B. Experiments Re-Design in 2011

Basically, EXP2011 followed the education approach described in Section III and the design of EXP2010 [3]. However, as EXP2010 contains several flaws and issues which may impede the effect of PSP education and affect the experiment results, in EXP2011 we made the several improvements to carry out a more rigorous and formal experiments. This section only describes the difference between EXP2010 and EXP2011 due to the limited space.

1) Selection of Students: In EXP2010, students were randomly selected to form pairs. Only those students (paired and solo) who finished all the assignments were selected for final analysis. Therefore, confounding factors of language skill and learning ability may affect the results. In EXP2011, we expect students with similar language skill as well as similar learning ability to be involved. Therefore, we referred to the scores of all the first year students in Java programming course and selected the candidates with scores in the range between 80 and 90 for EXP2011. By further applying statistic analysis of their scores in this range, the final range for selection locked between 82 and 87, which allows a total of 21 students selected. By consulting their overall performances in the Java course, these students’ comparable abilities in programming and learning were recognized and confirmed. According to the experiment design, these students were divided into two groups by the instructor (who did not know any one of these students): 7 students completing the PSP course individually, and 14 in pairs. This is based on the following considerations:

- Java is the only programming course in software engineering subject for freshmen, so the result of this course could reasonably reflect the student’s ability of programming. Since the size of observations is relatively small, we conducted Wilcoxon rank-sum test to the scores of selected students. The results show that the two groups have no significant difference in learning Java programming.

- The main purpose of PSP course is to help training qualified team members for software projects. Therefore, the training focuses on process disciplines instead of programming skills. The above selection criteria may help the students focus on process discipline in the experiment, and avoid spending much effort on struggling issues with programming language.

However, those students with higher score (over 90) were excluded because of the small sample size and their outstanding strength in learning from experience. Their performance difference between paired and solo styles in this course might be slim for observation.

2) Lab Environment: The two groups of students participating EXP2011 were provided the same lab facilities as other students, except that they were arranged into two dedicated labs.

3) Data Collection and Analysis: In EXP2010, we have discovered that paired students performed better than solo students in terms of their on-time submission rate of assignment and number of inconsistent items [3]. The former measure supports that paired students show higher involvement than solo students, while the latter indicates that paired students are able to follow PSP process discipline better than solo students. But in the final exam, there was no significant difference between these two forms of practicing. After analysis, we identified several potential issues in the original data measurement and analysis method. As an example, the original analysis only counts the number of those inconsistent items on the standard checklist. However, through an in-depth analysis of the checklist, we found that it’s not rare that several items come from the
same root cause. For instance, if a student fails to record the time log correctly, then he will get at least two inconsistent items, i.e., “time of the log is correct and reasonable” and “consistency on process data”. However, those two items are caused by the same mistake. It is more reasonable to combine the two inconsistencies as one. Another example is related to the analysis of the program quality. The original way considered defects data recorded in the PSP defect recording log only. However, since the data was recorded by the students themselves, it is hard to avoid data omission and cheating. As an improvement, we arranged TA testing all the programs submitted in EXP2011, and recording the number of residual defects into defects recording log for each submission. According to table V, we defined the procedures to collect data as the following:

**Inconsistent items** are collected from the evaluation results by TAs for each submission. Items from the same root cause are counted as one item and items occurred in submission and re-submissions for the same assignment are also counted as one item. TAs will collect and store the data.

*Program quality* is collected by TAs after they finished testing code from each submission. Only those defects discovered by TA are counted as residual defects.

*Exam scores* are collected by TAs after the final exam.

*Assignment durations* are collected by summarizing the actual time from time recording log for each submission. TAs calculate and collect the time data.

**V. ANALYSIS OF EXPERIMENT RESULTS**

To test the propositions in section IV, we conducted statistical analysis on the data collected in EXP2011 from four aspects, i.e., number of inconsistent items, program quality, exam scores and assignment duration respectively.

**A. Inconsistent Items**

As described above, we combined inconsistent items with the same source and used box-plot to show the distribution of the number of inconsistent items for both groups. Figure 1 depicts the comparison. Overall both paired students’ and solo students’ performance improvements can be observed. The numbers of inconsistent items in the last two assignments are lower than any before. It also shows that the paired students had fewer inconsistent items than solo students in the early of the course. When both groups turned to be more familiar with PSP process discipline, the numbers of inconsistent items for them tend to be closer and lower.

In order to characterize the status of inconsistent items in a clearer manner so as to guide future education, we categorized all the issues into ten types and summarized the number of inconsistent items for each type (shown in Table VI). All the inconsistent items discovered in the students’ submissions were exclusively assigned into their types.

Figure 2 shows the distribution of the inconsistent items across the types in descending order. The paired students had fewer inconsistent items than solo students in most types except Type 2 and 3. This confirms that paired students performed better than solo students on following PSP processes in most cases. It is also noted that Type 5, 1, and 8 constitute the most amounts of inconsistent items, which denotes both paired students and solo students made similar mistakes in assignments. The analysis result provides evidence for identifying improvements in future education. For instance, to address Type 5, more attention should be paid to explain the estimation methods in PSP and how to use the estimation form correctly.

**Finding 1**: Fewer inconsistent items are identified for paired students than that of the solo students in the early stage of PSP course.

**Finding 2**: Students lack enough skills to describe the issues, defects and corrective proposals.

**Finding 3**: Students may encounter challenges in using PROBE method and estimation form.

### TABLE VI. TYPE OF INCONSISTENT ITEMS IN SUBMISSIONS

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Insufficient or wrong description of defects on defect log</td>
</tr>
<tr>
<td>2</td>
<td>Time tracking issues (missing, inconsistent, wrong, etc.)</td>
</tr>
<tr>
<td>3</td>
<td>Wrong order of PSP phases</td>
</tr>
<tr>
<td>4</td>
<td>Incomplete implementation of the requirements</td>
</tr>
<tr>
<td>5</td>
<td>Wrong data on estimation form (plan and actual)</td>
</tr>
<tr>
<td>6</td>
<td>Inconsistent data (time, size, defect, etc.)</td>
</tr>
<tr>
<td>7</td>
<td>No test reports</td>
</tr>
<tr>
<td>8</td>
<td>Issues on PIP form</td>
</tr>
<tr>
<td>9</td>
<td>Did not follow coding standards</td>
</tr>
<tr>
<td>10</td>
<td>Wrong usage of review checklist</td>
</tr>
</tbody>
</table>

Figure 1. The number of inconsistent items for both groups

Figure 2. Summary of inconsistent items in types
B. Program Quality

To reflect the students’ program quality status more correctly, we measured the number of residual defects in the submitted programs in EXP2011 instead of the total number of defects in the recording log. Besides, we did not consider considering defect density when evaluating program quality because both groups received the same requirements in assignments and produced small-sized code. To do this, the TA group tested all submissions from both groups using a set of standard test cases developed for assignments. Figure 3 shows the detected defects in recording. There was zero defect in Assignment 1, 3, 4 and 6 for both groups. It might result from the low complexity and small size of Assignment 1 (calculating the mean value and standard deviation for a set of real numbers), and code reuse of Assignment in Assignment 3 and 4 plus Assignment 5 in 6. But in all the other assignments (2, 5, 7 and 8), where students had to produce new code, the total number of residual defects in the solo students’ submissions were higher than that of the paired students’.

Finding 4: In general, programs produced by paired students contain fewer residual defects than that of solo students in PSP course.

C. Exam Scores

The scores in the final exam can be one measurable indicator to reflect the effects of PSP education. To minimize subjective bias, we used multiple-choice questions in the final exam. The scores of both paired students and solo students are presented in Table VII. Results show that the paired students performed better than the solo students (64.21 vs. 58.14) in the exam. However, when further testing whether the difference is statistically significant, the scores of paired students tend to have larger values. We used SPSS to conduct a Wilcoxon rank-sum test on these two sample sets of independent observations due to the small size (7 sets of scores for both groups). The results are presented in tables VIII and IX, in which a p value of 0.043 (Asymp. Sig. (2-tailed)) is smaller than 0.05. It means that the paired students achieved significantly higher scores than the solo students. Finding 5: Paired students may perform better than solo students in terms of their exam scores in PSP course.

D. Assignment Duration

The assignment submissions contained time recording logs where actual time spent in each PSP phase was recorded. We calculated the total time spent on each assignment for both paired students and solo students. Since there were eight assignments, we collected 56 (>50) data points from each group. It enables us to use Levene’s test for equality of variances and t-test for equality of means for analysis. As the results shown in table X, for the first test, a significance value of 0.129 is much more than 0.05, which means there is no significant difference on the variances of two groups of data. For the second test, a significance value of 0.291 for equal mean assumption and a significance value of 0.293 for not equal mean assumption also indicate that there is no significant difference on the score between these two groups.

Finding 6: No significant difference is observed on the

---

**TABLE VII.** Score in final exam

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of data sets</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scores</td>
<td>Pair</td>
<td>14</td>
<td>64.21</td>
<td>6.375</td>
</tr>
<tr>
<td></td>
<td>Solo</td>
<td>7</td>
<td>58.14</td>
<td>5.367</td>
</tr>
</tbody>
</table>

**TABLE VIII.** Ranks

<table>
<thead>
<tr>
<th>Group</th>
<th>N</th>
<th>Mean Rank</th>
<th>Sum of Ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scores</td>
<td>Pair</td>
<td>14</td>
<td>12.93</td>
</tr>
<tr>
<td></td>
<td>Solo</td>
<td>7</td>
<td>7.14</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE IX.** TEST statistics

<table>
<thead>
<tr>
<th></th>
<th>Mann-Whitney U</th>
<th>Wilcoxon W</th>
<th>Z</th>
<th>Asymp. Sig (2-tailed)</th>
<th>Exact Sig.(2* (1-tailed))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scores</td>
<td>22.000</td>
<td>55.000</td>
<td>-2.024</td>
<td>.043</td>
<td>.046</td>
</tr>
</tbody>
</table>

---

**TABLE X.** Independent samples test

<table>
<thead>
<tr>
<th></th>
<th>Levene’s Test for Equality of Variances</th>
<th>t-test for Equality of Means</th>
<th>95% Confidence Interval of the Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>Sig.</td>
<td>t</td>
</tr>
<tr>
<td>Equal variances not assumed</td>
<td>1.061</td>
<td>60.612</td>
<td>.293</td>
</tr>
</tbody>
</table>
time spent on all the assignments between paired students and solo students.

E. Summary of the Result Analysis

The above results and analysis from EXP2011 support the four propositions described in Section IV: 1) there are fewer inconsistent items in submissions from paired students in early course stage, which means these students follow PSP process discipline better than solo students; 2) the submitted programs of paired students contain fewer defects, which means paired students’ program quality are better than that of solo students; 3) paired students have higher scores in the final exam than solo students; and 4) the time needed to complete the assignments is similar between paired students and solo students, which means there are no significant overhead to learn PSP by pairs. Besides, finding 2 and 3 reveal detailed information behind finding 1, which provides clues to improve future PSP education.

VI. DISCUSSION

The main contributions of this paper are two-fold: 1) proposed and implemented the improvements for the PSP education approach by adapting PP practices; 2) re-investigated the effects of this new PSP education approach with an improved experiment. Our findings from EXP2011 support that the new PSP education approach by paring can deliver better education effects than the standard PSP training suggested in [22]. This section discusses some issues related to both the education approach and the experiment for the considerations when applying this approach to other education scenarios.

A. PSP Training by Pairing

PSP features its personalization, i.e. self-customized software process and process data. Working in pairs in PSP course may affect this characteristic to some extent. However, the primary purpose of PSP training is to help students master the process related skills, such as project estimation, planning and tracking, product quality management etc., and also to prepare them for eventually working in teams (Team Software Process [23]). Therefore, personal process and process data at the learning stage are not necessary to be as the same as that in real projects.

1) Self-customized Software Process:

Students need to understand the software process before they can define their own. In the education approach proposed in this work, paired students have no difference from solo students in lectures. In the lab session, the driver in a pair practices PSP process almost in the same way as a solo student. Although the chances (number of assignments) for this kind of practice drop by half, they are provided the opportunities to observe and correct their partners’ process discipline in the rest of the assignments. In short, the observers can also learn by observing. The value of this role changing was reflected in EXP2011. Our results indicate that paired students can perform better than solo students in understanding and mastering the PSP process discipline in this course.

2) Self-customized Process Data:

Although the process data collected during PSP course is important to establish students’ performance baseline in PSP course, it offers limited reference value in real world software projects due to the big gap between student exercises and industrial projects. In this course, students can understand the practices and process in PSP and master the skills to manage personal process. They may start to establish and maintain their customized process data through future real projects.

B. Adapting the Approach to Regular Semester

The PSP course described in this paper takes ten continuous days. In academic environment, however, a typical course usually spans over one semester. The biggest challenge in stretching this course design to fit regular semester setting is how to schedule the sessions in a weekly manner. The current course consists of a number of three hour sessions, which can be scattered into each week of the semester. The lecture and lab sessions can be interchangeable. Considering the mean time of each assignment is around 150 minutes, lab session can be provided with multiple batches with a smaller number of students in each. This arrangement may allow more space and further mitigate mutual interference. Hence, the full course can be completed in ten weeks, which can fit into most semesters.

C. Suggestions to Delivery PSP Course

To establish and maintain process discipline at individual level is one of the most important objectives for PSP course. Based on the findings from EXP2010 and EXP2011, we proposed several suggestions which can improve the delivery of PSP course.

1) Focus on PROBE Method and Estimating Form:

PROBE method is used to estimate size and time in PSP assignments. However, application of PROBE method is difficult for most novices. Students often get confused about the four different types of PROBE method with different historical data criteria. Therefore, PSP instructors should put more time and attention to explain the PROBE method together with the usage of estimating forms. For instance, more examples may help the understanding of the different criteria for historical data when applying PROBE.

2) Attention on Description of Defects and Process Improvement Proposals (PIP):

Insufficient descriptions about the defects and PIPs are also a common issues occurred in students’ submissions. In defect logs, students tend to describe the symptom of defects rather than the root causes. While the symptom creates nearly no value for future defect prevention, this kind of description provides very limited help for students. Similarly, in PIP form, students are expected to show insights and understanding about their own processes. Therefore, descriptions in the PIP form should contain analysis results from process data. For example, PSP instructors may encourage students to seek reasons why they achieve better or worse estimation, quality, productivity and so on.
3) **Strict Evaluation of Early Submissions:** To deliver PSP course, instructors often follow a step-by-step strategy, i.e. deliver PSP by incremental levels, then correct students’ development behavior gradually. However, it is not easy to make these changes. The evaluation remarks from TAs play a vital role to tune students’ personal process. Hence, both instructors and TAs should be extremely strict in early evaluations and point out as many as possible inconsistent items. This can force to establish the correct process discipline and habits, and relieve the pressure in evaluating late submissions.

4) **Ratio of TAs to Students:** Since TAs should provide instructive evaluations to submissions, more information need to be included in the evaluation. As a consequence, it increases the evaluation time on single submission. Therefore, to provide timely evaluation to the daily submissions, the ratio of the number of TAs to students became a crucial factor. According to time measurement results, each TA spent on average about 30 minutes on evaluating one submission in EXP2011. Therefore, no more than ten submissions can be allocated to one TA group each day. However, with proper training and practice, the number of the TAs can be greatly reduced by allocating more submissions to each TA group.

**VII. Threats to Validity**

Aiming to continuously demonstrate, evaluate and improve the suggested PSP education approach by adapting PP practices, we conducted the second experiment (EXP2011) with the improvements in both the PSP education approach and our experiment design.

In order to test the propositions in EXP2011, we collected the data in terms of the metrics suggested in Table I. Most of the metrics were based on the evaluation of students’ assignments or exam, where one major threat to the validity of this study might exist. As all evaluations were done by TAs in this course, their subjective bias in evaluations should be considered and controlled in this experiment.

The submitted assignment documents contained students’ names, which did not support the blinding criteria suggested in most experiments. All TAs in EXP2011 were asked to strictly follow a standard checklist and marking guide that were provided by the instructor in finding the inconsistent items. The use of checklist and guide mitigated the potential evaluation bias due to the exposure of student’s name.

When evaluating program quality, TAs were only provided with the submitted source code but without knowing students’ names and groups (paired or solo). In addition, to detect residual defects they ran a set of test cases that were developed according to assignment requirements in advance, which also minimized the possible bias in program quality evaluation.

In the course’s final exam, all students were tested with multiple-choice questions only. However, multiple-choice questions may have limited power to show education effects. According to Bloom’s taxonomy of educational objectives [24, 25], multiple-choice questions are more suitable to test low level skills in the cognitive domain, e.g., knowledge and concepts of PSP. We may seek more effective methods in evaluating education effects but keep potential bias well-controlled.

Another threat to validity may come from the time log recording. The data of assignment duration were calculated from the time-logs that were recorded by the students themselves. Although there might be mistakes in time recording, we consider it is also common even in industrial settings. However, to minimize the number of mistakes in recording, in the early lab sessions, the PSP instructor worked on-site and provided guidance to help students establish good habits to track and record time data.

Last but not least, in order to make the experiment results generate in most academic education environment, EXP2011 also managed to mitigate the possible impact from the deviation of students’ personal skills or capabilities. We only selected the samples with similar programming skill and learning capability from the enrolled students, and randomly assigned them into paired or solo group. This is also one important improvement to EXP2010 design.

**VIII. Conclusion**

The rapidly growing IT industry in China creates an increasing demand of software professionals. Personal Software Process (PSP), as widely-accepted process training for software practitioners, can equip students with tactical development skills that the industry needs. However, the success of PSP education in academic environments needs carefully-crafted approaches to tackle the challenges, such as large class size and students’ involvement. This paper, on the top of our prior experiment reported in [3], proposes an improved approach to adapting typical practices of Pair Programming (PP) into PSP education. This approach aims to significantly relieve the workload on assignment evaluation, to increase the quality of students’ work products and their involvement, as well as to improve the education effects.

The paper also reports a follow-up experiment and results that further tested the improved PSP education approach. With reference to the experience and feedbacks from EXP2010, a new experiment (EXP2011) with more rigorous designing and controlling was implemented. The analysis of the experiment data and our findings support all the propositions, and confirm the positive impact of adopting collaborative learning in PSP education.

We have proposed, implemented, and improved a new approach to delivering PSP course in academic environments by adapting PP practices. We have also investigated this approach using two experiments, which all confirmed its value on PSP education. Our research and experience at this stage also raise a few interesting topics for future research, such as

- How to comprehensively validate the effect of the PSP education approach proposed in this work? The measurement and analysis conducted in EXP2010 and EXP2011 focus only on the learning stage of PSP,
However, the objective of PSP education is to prepare team members for TSP teams. It’s necessary to further validate the education effect within TSP teams, especially for those students who completed the PSP course in pairs.

- How to generalize the findings identified in this experiment to regular PSP courses. The students involved in EXP2011 were selected to have nearly identical programming skills; however, a regular class may form with students with various language skills and learning ability. Therefore, it’s necessary to investigate the impacts and limits of applying our approach to these students.

- How to apply and improve this education approach into industrial environments? The approach has showed its value in academic education environment. However, in industrial settings, PSP training may encounter different challenges, such as the large number of software practitioners in China. It may be more difficult to find a sufficient number of TAs required in industry training than in universities.

We encourage other SE educators to continuously experiment, validate, adopt, and improve this approach in their PSP education practice, and report their experiences.

ACKNOWLEDGMENT

This work was partially supported by the Fundamental Research Funds for the Central Universities, Project of Software Engineering Specialty Construction (2007-2201-2203), the Project of Software Engineering Teaching Team (2010-118), and the Project of Software Engineer Education Innovation (2008-042) by the Ministry of Education of China.

Corresponding author: Dong Shao (Email: dongshao@software.nju.edu.cn)

REFERENCES


[22] SEI PSP course material version 2006.


[24] Bloom, Benjamin S. Taxonomy of Educational Objectives (1956), Published by Allyn and Bacon, Boston, MA. Copyright (c) 1984 by Pearson Education.