Peer Instruction in Online Synchronous Software Engineering – Findings from fine-grained clicker data

Bhuvana Gopal
Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, USA
bhuvana.gopal@unl.edu

Stephen Cooper
Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, USA
stephen.cooper@unl.edu

***Abstract*—In this Research Full paper, we present the results of a replication study in a semester-long, sophomore-level software engineering course utilizing Peer Instruction (PI). PI is an active learning pedagogy with roots in STEM Education. In this study, we examine the relationship between student response data from in-class PI correctness and students' performance on quizzes and exams. We worked with a fully remote, synchronous course offered over Zoom. The study we replicated was with an honors cohort of students with a diversity of undergraduate majors, while we focused on a non-honors course containing computing-related majors.**

Our intervention design included a flipped-classroom approach for each class session with required readings, reading quizzes, followed by PI in class using online breakout rooms for peer discussion. Our course modules were heavily based on industry practices and knowledge from the workforce, across several varied modules that encompass the complete software development lifecycle, and were as follows: Software Process Models (SPM), Software Architecture (SA), Databases (DB), User Interface/user Experience (UI/UX), Software Testing (ST), and Continuous Integration (CI). Our data points for analysis with fine-grained PI student response data were two-fold: scores from weekly online quizzes, and a summative final exam, administered online through a course management system (CMS), at different points during the semester after the PI sessions. The online quizzes and the online exam were timed, closed book/notes, and conducted during class periods.

We analyzed and classified individual student responses before and after each question in each module and attempted to create response patterns for each module. We correlated these response patterns with exam and quiz scores using ANOVA techniques, on a variety of questions including Parson's problems. We report overall correctness on each type of vote, track student response patterns from in-class to quizzes and the exam, and quantify absolute percentages of students that demonstrate longer-term learning from the PI process.

Our results show that 58% of students exhibited cognitive gains across all modules during PI sessions. Students who learn in class from PI perform well on the quizzes and the final exam, indicating persistence of the knowledge gained during PI several weeks after the actual sessions. We also found that those who fail to learn from the PI process in the class perform worse on quizzes and the final exam. Our results were consistent across all modules. More significantly, we found PI to be an effective way to teach our software engineering course based on student learning before and after PI, in a completely virtual environment, a result unique to our study. Based on our results, we discuss the implications for software engineering education, both in-person and virtual.

***Index Terms*—Software Engineering Education; Peer Instruction; Active Learning; Online courses; Virtual Instruction; Cognitive gains using Peer Instruction; Active Learning; Fine-grained data analysis; student response data**

I. Introduction

Over the last few years, a significant body of research has emerged, attesting to the effectiveness of Peer Instruction (PI) as a pedagogical approach for teaching CS courses. PI encourages student discussion and active involvement, and engages the class in the discussion and analysis of multiple-choice questions [1].

The extent to which students learn from discussions resulting from in-class multiple choice questions and the extent to which this learning manifests on a final exam have been studied in several introductory CS courses [2]–[6]. Two of these studies correlate the increase in in-class correctness due to PI to the single final exam [3], [6] to find statistically significant relationships. More recently, Gopal and Cooper [7] examined possible relationships between in-class student answer correctness, and quiz and exam performance in an undergraduate software engineering course. Their work was based on in-person instruction. Their student population comprised of only honors students, belonging to a variety of majors such as CS, Business, Finance, Actuarial Science and various Engineering disciplines.

The present work seeks to replicate the study by Gopal and Cooper [7] and extend it in the context of a fully synchronous online sophomore software engineering course. While they conducted their study on honors students from various academic majors, we conducted this study on primarily computing related majors (Computer Science and Computer Engineering) who were not honors students. Similar to their study [7], we utilized multiple data points on quizzes and an exam administered weeks apart from each other for our analysis. We used PI questions in class to measure what students learn from their peers and the instructor, across six

individual modules, each module corresponding to a software engineering module in our course. We also studied the same six modules [7], namely, Software Process Models (SPM), Software Architecture (SA), User Interfaces/User Experience (UI/UX), Databases (DB), Software Testing (ST) and Continuous Integration (CI). We then included two other isomorphic (same concept) versions of the PI question on a module-wise quiz and an exam to measure longer-term retention of learning across several weeks.

The rest of the paper is organized as follows. Section II describes the related literature. Section III details the research methods that we used in this study. Section IV presents our results, Section V discusses and analyzes those results, Section VI presents possible threats to the validity of our study, and Section VII concludes.

II. Related Literature

A. Active Learning and Evidence Based Instruction

Student learning and attitudes are both enhanced by the use of evidence based instructional practices [8], [9]. Active learning is a collection of teaching approaches that places the student at the center of their learning, instead of accepting them as passive listeners [7]. Some examples of active learning pedagogies are structured team-based (cooperative) learning [10], paired discussions, journal writing, problem solving in groups, and even case studies and role playing. The advantages of active and cooperative learning have been well documented [11]. Pair programming and other active and collaborative learning techniques outside of traditional lecturing have been used by computer science educators for a while [12], [13]. The concept of a "flipped classroom" or "inverted classroom" is explained by Maher et al. [14]. Such a classroom involves asking students to prepare for lecture in multiple ways such as a reading assignment, watching a video, or exploring programming techniques [15].

B. Peer Instruction – An introduction

The active learning practice, Peer Instruction (PI), was originally developed by Mazur [16] in the early 1990s for introductory calculus-based physics for non-majors. PI is an evidence-based pedagogical practice where students are asked conceptual questions during class time before and after they discuss the possible answers with their peers, and their answers are collected in real time, through student response systems known as clickers [17], or response cards.

C. Peer Instruction in CS

There is a growing body of literature regarding PI in CS [2], [18]–[20]. Studies have shown that PI is well-liked by both students and instructors [1], [19], [21]. Studies have shown that students learn from the discussion phase when isomorphic (same concept, differently worded) questions are used [2], [19], [22], [23]. In computer science, PI has been shown to be valued by students as well as instructors [2], [24]–[26], and result in individual learning [21], [25]. In addition, PI has been shown to improve student efficacy [24] as well as reduce fail rates [26], and help in identifying struggling students early [4].

Most of the above studies using PI in CS have been in introductory programming courses. However, PI is making inroads into upper division computing courses such as computer architecture and theory of computing as well as in cybersecurity [18], [27]. In software engineering, PI has been studied in the context of student discussions [28] and risk management [29], [30]. More recently, PI has been studied in software testing [7].

D. Study Replication

While there are studies exploring the relationship between PI and student attitudes, few studies have focused on the relationship between student responses from PI and student learning outcomes in exams. One study by Zingaro and Porter [6] found that clicker question performance during the term relates to performance on the final exam. They used isomorphic questions for PI and per-class clicker data to track students as the students progress through the term, and binned students by quartiles in the first weeks of the term. They found that those bins remain consistent relative to one another throughout the term and, ultimately, on the exam. They also found that students who learn in class through PI perform very well on the final exam. These students perform as well as those who understood concepts prior to the classes in which those concepts were taught. Gopal and Cooper [7] adopted PI in a sophomore software engineering course, classifying students into correctness bins based on PI session answer correctness in class. Their student population consisted of a cohort of honors students, belonging to a variety of academic majors. Their study has shown that students who learn through PI learn as well as students who knew the content to being with, and that the cognitive learning gains from PI persisted through several weeks after the PI session [7]. In this paper we discuss our attempt to replicate the study by Gopal and Cooper [7] in an online, synchronous software engineering classroom with non-honors students belonging to computing majors such as Computer Science and Computer Engineering.

III. Methods

A. Study Context

In this study, we targeted students enrolled in a semester long (15 week) sophomore level software engineering course at a large R1 university. The entire course was conducted synchronously online through a remote video-conferencing tool, Zoom [31].

In the original study by Gopal and Cooper [7], the student population belonged to a variety of majors - CS, CE, SE, Actuarial Science, Marketing, Finance, Electrical Engineering and Business. In our study, there were 55 students in the class; 51 of them were CS majors, and 4 students were Computer Engineering majors. Our implementation of PI during this course was based on [2], [21] and [7], following the steps outlined below:

(a) **Before class**: Students were assigned preparatory readings, to learn some of the basic concepts, or definitions. This approach allowed for preliminary reading material to not be presented in class, facilitating the creation of time during class for student engagement. A graded short online quiz was given before each class period before the instruction started.

(b) **During class**: The instructor delivers a short lecture over Zoom. At multiple points during the lecture, students engage with questions designed to help them confront and explore challenging concepts [19]. Multiple-choice questions were built into the lecture content, and students gain participatory credit for answering the questions with a student response system, also known as a clicker [32]. These questions were administered on the clicker system's website [17]. Timed polls were given for each multiple choice question, for each vote.

The specific sequence of steps for a clicker question episode in class was as follows:

(i) **Initial vote (Q1)**: Question posed, and students answered individually (results not displayed for class). The correctness threshold of responses in this initial response to warrant a small group discussion generally varied from 30 to 70%, as per Mazur's guidelines [16].

(ii) **Zoom breakout room discussion in small groups (3-5 students)**: Students discussed their analyses over Zoom and shared their reasoning with each other, perhaps convincing their Zoom groupmate(s) to change their responses for the second vote. This discussion took place in pre-assigned breakout rooms where each student had 2-4 other students to discuss their reasoning with. At the beginning of the semester, students were grouped randomly and assigned to breakout rooms. These breakout room assignments were kept consistent throughout the semester, for ease of tracking. For example, if student A had been assigned to breakout room 3, they would remain in breakout room 3 for the entire semester, for every single PI question in class.

(iii) **Second vote (Q2)**: Students answered a second time, perhaps changing their answer based on group discussion.

(iv) **Typical duration of each vote**: Depending on the level of difficulty of the PI question as perceived by the group, breakout room discussions would last anywhere between 45 seconds to 2.5 minutes.

(v) **Instructor led discussion**: After Q2, the instructor would lead a class-wide discussion by asking students to share explanations and discussions they had in their group and providing clarification of how the question can be analyzed. The correct answer was clearly indicated. The results of student responses were displayed at this point. We put forth significant effort in the classroom to explain to students why PI was being used and why it could be beneficial for them, following suggestions by Simon et al. [4]. In our implementation of PI, unlike Zingaro and Porter [6], we did not conduct a vote after the instructor intervened, since our aim was to study how peer discussion influenced student learning gains. This is consistent with the PI implementation utilized by [7].

Our aim was to study how students gained from peer discussion. We use the following terminology for our data points:

- **Q1**: The individual vote on the first (in-class) question.
- **Q2**: The individual vote on the second (in-class) question, which occurs after peer discussion.

We collected clicker data from six modules that we have taught during the semester.

- Module 1 (UI/UX) - UI/UX design
- Module 2 (SPM) - Software Process Models
- Module 3 (SA) - Software Architecture
- Module 4 (DB) - Database design
- Module 5 (ST) - Software Testing
- Module 6 (CI) - Continuous Integration

B. A typical day in the PI classroom - an example

Let us consider a typical day in our classroom from the perspective of a fictional student named Mary. Let us assume that the module meant to be taught in this example class session is Databases (DB). A day before the class starts, the instructor posts a set of reading links on Canvas [33] under the DB module and makes an online announcement. Mary comes into class having read through the content in these links. As soon as class starts, a short, five minute online reading quiz is administered on Canvas. Mary also signs into the online clicker system at this point, using their personal login, to get ready for PI questions in class. Once the quiz window closes, the instructor starts with a short lecture on database related concepts. A few minutes in, the instructor poses a PI question, and opens up the poll for the first vote (Q1) by posting the question online [17]. An example question would be:

Which is the most suitable type of database for storing sensitive data?

(i) RDBMS
(ii) XML
(iii) NoSQL
(iv) Open source
(v) Only (i) and (ii)

The correct answer to this question is option (i) RDBMS. Students answer the question within the allotted time frame of 45 seconds to 1.5 minutes, as do all other students in the class. Mary answered option (ii) XML. The instructor closes the poll, quickly examines the statistics on what percentage of students got the answer right, and if more than 70% of the class answered correctly, proceeds on with the rest of the lecture. In this example scenario, less than 70% answered correctly; the instructor opens up pre-assigned breakout rooms on Zoom.

There are 55 students in the class, and a total of 11 breakout rooms, with 3-5 students each. Mary was previously assigned

to breakout room 3 when the course began a few weeks ago, and will remain in the same room. Students gather in the breakout room, and discuss the PI question, which they just voted for. At this point in time, none of the students are made aware of how the class voted for Q1, or what the correct answer is. After a time frame of 1.5 minutes, breakout rooms are closed and students return to the main class Zoom session. The instructor now opens up the poll (Q2) on the iClicker website portal [17]. Mary and her fellow students proceed to vote again, this time with newly minted ideas from their peer discussion. After 2 minutes, the instructor closes the poll, and students return to the main Zoom session again. This time, Mary answered (i) RDBMS. The instructor now explains the correct answer and the distractors, and offers clarifications if any. The instructor resumes lecture, and the cycle repeats. Throughout the session, lecture and PI questions alternate, and by the end of the session, Mary would have answered Q1 and Q2 (if Q2 was offered), for all 5 of the PI questions in the Databases lecture.

C. Research Questions

We sought to study fine-grained clicker data by classifying individual student responses before and after each question in each module and attempted to create response patterns for each module using PI questions, available in the original study by Gopal and Cooper [7]. We correlated these patterns with exam and quiz scores, as a means of measuring cognitive retention across the duration of the course. We applied this methodology to a variety of questions including Parson's problems [34].

In this study, we report overall correctness on each type of vote, track student response patterns from in-class to the exam, and quantify absolute percentages of students that demonstrate longer-term learning from the PI process.

We wanted to understand how the correctness of student responses during online PI sessions correlated with their quiz and exam scores. Our focus was not on how many questions each student answered correctly during an online PI session, but rather how the overall performance trend of each student in each PI module correlates with that student's performance on quizzes and the exam. Specifically, we aimed to address the following research questions.

- RQ1: Did PI help students learn during each online session for each module? Specifically, did PI help non-honors students belonging to computing majors learn during each online session for each module?
- RQ2: In each module, how well do students' response patterns in individual online PI questions correlate with online quiz and exam scores? For example, if a student obtained mostly incorrect responses before PI discussion and mostly correct responses after PI discussion, were they likely to do better on the quizzes/exam pertaining to that module than students who did not correctly answer after PI discussion?

D. Intervention Design

This research project was determined to be exempt research by our school's Institutional Review Board. The course consisted of two 75-minute online lectures delivered synchronously each week over 16 weeks. We utilized a flipped classroom approach [14] in our course with students having to complete required readings before each class session. Similar to Gopal and Cooper [7] we utilized reading quizzes [6], where students complete a small amount of reading and respond to several questions before each lecture. We administered these quizzes online through Canvas. Reading quizzes as well clicker responses were graded based on completion and not correctness.

Each module in the course consisted of an in-class component within which the PI questions were embedded. Each in-class session was followed up with a take home lab, also administered online on the CMS, supported by TAs. Nearly a week after each module was completed, students answered an online, timed, closed book/notes quiz on the module. Approximately one month after the completion of the quizzes, students completed an online, timed, closed book/notes written exam containing questions from these modules and concepts taught using PI. 55 students completed 8 assignments, 6 quizzes (one on each module), and the written exam. The assignments students completed provide a richer data stream to analyze. Grading them is more nuanced and beyond the scope of this study. For this study, we focused on the data from the quizzes and the exam.

E. Instrument Design

In the original implementation of PI, Mazur [16] utilized conceptual multiple-choice questions called ConcepTests that were originally designed for students in large physics classes. ConcepTests typically focus on a single concept, cannot be solved using equations, have good multiple-choice answers, are clearly worded and are of intermediate difficulty. Zingaro and Porter [6] utilized isomorphic questions, which were differently worded questions for each concept. While this fits well with introductory CS courses such as the one on which they conducted their study, we chose to utilize the same PI question before and after PI discussions, since software engineering modules facilitate more open-ended questions than introductory CS courses [35]. We utilized PI questions formulated by [7] to maximize the extent to which questions were clear, concise, and well-targeted for our intended group.

The instruments we used to collect data were three-fold: first, we utilized student response scores for the online PI questions on the course modules outlined in Section IIIA. Second, we collected data from student responses in the online, module-wise, closed-book quizzes two weeks from the class session on each module. Our final instrument was the student response data from the online, summative, closed book exam which covered all the modules, which was conducted four weeks after the last PI module presented in class. While each quiz pertained to one specific module, the exam contained questions from all six modules. We culled out the scores for

each individual module from the exam and utilized them as our third data point.

F. Classifying student response patterns into correctness bins

Similar to Gopal and Cooper [7], we utilized fine grained clicker data for each individual student from the SRS [36] to analyze response patterns for each student for every single question in each module (W - wrong, R - right). We analyzed data for both votes, Q1 and Q2 per question per student. We grouped per PI question responses into four bins as follows, following the grouping used by the original study [7]:

(a) R-R: This represents a student who got Q1 and Q2 correct; they arrived at the correct answer before peer discussion happened and maintained their response correctness after the discussion.
(b) W-R: This represents a student who got Q1 incorrect and Q2 correct. They arrived at the correct answer after peer discussion.
(c) W-W: This represents a student who got Q1 and Q2 both incorrect. They started out with an incorrect answer and remained with an incorrect answer choice after peer discussion.
(d) R-W: This would denote that a student got Q1 correct, and after peer discussion, picked an incorrect choice.

Once we categorized each individual student response for each PI question into one of these four bins, we determined an aggregate response pattern for each module based on the majority response bin for that student for that module. If a student had, for example, W-R for 5 out of 7 questions in a given module, then, they would be a W-R for that module. Table I shows the various modules and the correctness criteria for each module, in terms of number of questions. Column 2 shows the total number of questions in each topic, while Column 3 shows the minimum number of questions a student must answer correctly in each of those topics, for the student response to be classified into a bin.

TABLE I
MODULE-WISE CORRECTNESS CRITERIA

Topic	Total number	Correctness criterion
SA	2	2
SPM	3	3
DB	5	4
UI/UX	1	1
ST	7	6
CI	3	3

In Zingaro's study [6], there was an additional vote, Q3, which was taken after the instructor explained the correct answer. They traced student response patterns on Q1, Q2 and Q3 using a decision tree. There is also the notion of a Control Group and Potential Learner Group, the terminology used by Porter et al. [37], which translate into the R-R and W-R bins respectively, described as follows:

- Control Group (CG): the group of students who correctly answer Q1 and correctly answer Q2. This group corresponds to the R-R group in our study.
- Potential Learner Group (PLG): the group of students who incorrectly answer Q1 and then correctly answer Q2. This group reflects the most obvious subgroup of students that may have learned from the PI process. This group is represented by the W-R bin in our study.

It is worth noting here that while Zingaro and Porter [6] utilized the CG and PLG, Gopal and Cooper [7] utilized the terminology R-R and W-R groups respectively. We follow the same terminology guidelines set by Gopal and Cooper [7].

IV. RESULTS

A. Statistical Analysis

We analyzed a rich set of data from three different data sets each collected a few weeks apart. To correspond with our three research questions given earlier, we report overall correctness on each type of vote (Q1 and Q2), track student response patterns from in-class to the exam, and quantify absolute percentages of students that demonstrate longer-term learning from the PI process.

To test whether specific groups of students benefited from PI, we analyzed each combination of bins for each module separately. For each of the 6 topics, we analyzed W-R vs R-R and the W-R vs W-W bins, once for the topic-wise quiz, and once for the exam. We used the Shapiro-Wilk test to see if the data was normally distributed. The test combines skew and kurtosis to produce an omnibus test of normality. We checked for homoscedasticity using Levene's test for homogeneity of variance across all values of the independent variables. Based on the results of these tests, we chose the appropriate ANOVA test (regular or Welch's) with associated p-value to determine the statistical significance of the results [38]. Throughout this paper, we use a p-value of less than 0.05 to indicate that the results found are significant.

B. Overall correctness during PI sessions

Table II shows the percent of students in each bin for each module.

TABLE II
MODULE-WISE BREAKDOWN OF PERCENT CORRECT IN STUDENT RESPONSE PATTERNS

%	UI/UX	SPM	SA	DB	ST	CI	Avg.
W-R	56	58	62	65	51	58	58
W-W	0	5	7	15	9	22	10
R-R	44	37	31	20	40	23	32
R-W	0	0	0	0	0	0	0

Table III shows the Module-wise breakdown of W-R as a percentage of (W-R+W-W):

A significant but expected result is that we found no students in the R-W group, indicating that no one who was right to begin with in answering Q1 changed their mind after peer discussion to answer Q2 incorrectly. We found that on average,

TABLE III
MODULE-WISE BREAKDOWN OF W-R AS A PERCENTAGE OF (W-R+W-W)

UI/UX	SPM	SA	DB	ST	CI	Avg.
100	92	90	81	85	73	87

68% of all students initially got Q1 wrong (Table II, Average of W-R + Average W-W); 87% of them answered Q2 correctly (Table III). The module-wise breakdown for these figures can be found in Tables II and III. PI seems to have positively impacted almost all of the students in our sample, to change their initially incorrect answers to correct answers. For two of the modules (DB and CI), there was a relatively large percentage of students who fell into the W-W category (15% and 22% respectively). There were no W-W students for the UI/UX module (Table III). This denotes the group for which PI probably had no effect on students. Approximately a tenth of the student sample (10%) fell into this category (Table II).

PI did not seem to derail students in the R-R group. The two main groups that we are concerned with are the W-R group and the R-R group, and we will discuss the significance of these results in Section V.

C. Correlation of student response bins with quiz scores

As Table II indicates, we found 3 groups of students – W-R, R-R and W-W. We wanted to investigate if the students who got Q1 incorrect and Q2 correct, (the W-R bin) maintained their correctness in quiz and exam scores. We also wished to study if students in the W-R bin performed any differently in the quizzes and exam than the students in the R-R group or W-W group. Did the W-R group maintain their answer correctness on quizzes and the exam? Did PI correctness translate to quiz correctness and/or exam correctness? These were some of the questions we sought answers to with our data. Tables IV and V show the raw average percent scores for quizzes and the exam for each of the 3 PI bins/groups.

TABLE IV
BIN-WISE RAW AVERAGE PERCENTAGE QUIZ SCORES FOR EACH MODULE

	UI/UX	SPM	SA	DB	ST	CI
W-R	76	82	81	78	79	71
R-R	74	83	82	80	82	73
W-W	-	22	26	20	21	24

We correlated the W-R, R-R and W-W bin scores with the quiz scores across the modules using one-way three variable ANOVA tests with post-hoc pairwise t-tests. We found that there was no statistically significant difference between the W-R bin or the R-R bin, across the modules for the quizzes and exam. We also found that there was a statistically significant difference in the scores across modules for the quizzes and exam, between the W-R and W-W bins.

TABLE V
BIN-WISE RAW AVERAGE PERCENTAGE EXAM SCORES FOR EACH MODULE

	UI/UX	SPM	SA	DB	ST	CI
W-R	60	50	83	70	71	75
R-R	50	60	76	72	72	80
W-W	-	17	23	19	24	22

Table VI shows the ANOVA p values across each module for quiz scores in correlation with the W-R and R-R bins, and the post-hoc tests for the W-R bin vs the W-W bin. We found that the W-R group had statistically significant differences in their quiz scores compared to the W-W group in every module.

TABLE VI
ANOVA P VALUES FOR W-R, R-R AND W-W IN MODULE WISE QUIZZES

Quiz	UI/UX	SPM	SA	DB	ST	CI
W-R vs R-R	0.96	0.51	0.92	0.95	0.96	0.68
W-R vs W-W	N/A	0.001	0.003	0.001	0.001	<0.001
R-R vs W-W	N/A	0.001	0.001	0.001	0.001	<0.001

D. Correlation of student response bins with exam scores

We correlated the W-R, R-R and W-W bin scores with the exam scores across the modules using one-way three variable ANOVA tests with post-hoc pairwise t-tests where applicable, and again found that there was no statistically significant difference between the W-R and R-R bins. Again, we found that there was a statistically significant difference in the performance of the W-R bin compared to the W-W bin on the exam questions in those modules. Table VII shows the ANOVA p values across each module for exam scores in correlation with the W-R and R-R bins, and the post-hoc tests for the W-R bin vs the W-W bin.

TABLE VII
ANOVA P VALUES FOR W-R, R-R AND W-W CORRELATED WITH EXAM SCORES IN EACH MODULE

Exam	UI/UX	SPM	SA	DB	ST	CI
W-R vs R-R	0.95	0.41	0.87	0.94	0.98	0.70
W-R vs W-W	N/A	0.001	0.002	0.001	0.001	<0.001
R-R vs W-W	N/A	0.001	0.001	0.001	0.001	<0.001

V. DISCUSSION

We find consistent agreement with our Q1 and Q2 scores within each PI session with results from previous studies [2], [5], particularly with the study by Gopal and Cooper [7] which we have replicated in our work. Our results indicate that almost all students revised their initially incorrect answers to obtain correct answers after PI was administered (Table II). This is potentially significant in the light of another noteworthy finding in our study - that we had no students in the R-W

group; this indicates that PI did not negatively impact the decision-making process for answer correctness.

Zingaro and Porter [6] found raw learning gains of 25% and 21% within the PI session and with exam scores respectively. Gopal and Cooper [7] reported that 54% belonged to the W-R group. We obtained similar results with 58% of all students answered Q1 incorrectly and Q2 correctly, as evidenced by the W-R group across all modules (Table II) within PI sessions. Our analysis confirmed that there is a positive correlation between the in-class PI session performances of the W-R group with module-wise quiz scores (Table VI) and exam scores (Table VII).

A significant observation from our study is that over a course of a semester, students who understood software engineering concepts before PI (R-R), as well as the learners who gained from PI (W-R), had no statistical difference in their quiz scores and exam scores. While we found that most of what students learn during the PI cycle is observable on the quizzes and the final exam, students in the W-R group also performed consistently better than the students that stayed W-W during the PI cycle (Table VI, Table VII), and that the learning gained from PI persisted over several weeks.

Students either initially answered Q1 correctly, maintaining their correctness through Q2 (R-R group), or students answered Q1 incorrectly, learned from PI, and corrected their answers for Q2 (W-R). Gopal and Cooper [7] had two modules with students in the W-W group; we had 5. The percentage of students in the W-W group was relatively small across all modules. The W-W average (10%) was relatively much smaller than the W-R average (58%). Much larger numbers of students learned from PI as opposed to those who did not.

Similar to the study by Gopal and Cooper [7], we found no statistical difference in the overall performance of the W-R and R-R groups, either in the quizzes (Table VI) or the exam (Table VII). This is also consistent with the results by Zingaro and Porter [6] whose PLG and CG corresponded to the W-R and R-R groups respectively.

We found maximum percentages of students go from incorrect to correct responses in the SA and DB modules, followed by SPM, UI/UX, ST and CI. Based on our results in Table II, we saw that a large proportion of students who gave incorrect initial responses obtained correct responses after PI (87%) (Table III). Based on the results of our study, we now answer our research questions as follows:

- **RQ1:** PI clearly helped students learn in in each module. In analyzing the response patterns of the W-R group, we found consistently large numbers of students go from incorrect to correct answer after PI discussion, with nearly all the students who initially answered Q1 incorrectly answering Q2 correctly.
- **RQ2:** Students in the W-R group, which is the group that potentially benefited from PI directly, did at least as well as the R-R group in the quizzes for every module and exam (Tables VI and VII). They also performed significantly higher than the students in the W-W group. Students who learned from PI retained their learning for all modules through the course of the semester.

VI. Threats to validity

In our study, several weeks or more pass between the in-class questions, the quizzes and the exam. This time lapse, while giving us the ability to investigate learning outcomes over a period of time, hinders our ability to tightly control what students can learn from one setup to the next. Similar to the study by Gopal and Cooper [7], this has at least two implications. Some students may revisit Q1 and Q2, or may thoroughly study the associated topics, while others may not. There is also the effect of lab assignments and homework assignments on learning, which we have not investigated in this study. Despite this limitation, our data suggests that PI itself predicts quiz and exam scores in an online synchronous course. Relative to the students in the W-R group, those who respond incorrectly to Q2 (W-W group) perform significantly lower on quizzes and the exam. These Q2-incorrect students may have indeed studied the relevant topics again prior to the exam, but this evidently did not make up for failing to learn the material during the PI process. What remains unclear is why the W-W group failed to learn from the peer discussion; however, this is a relatively small number of students. It would be interesting to study whether this group of students would be larger in the absence of PI, and with a larger student sample.

Second, our in-class questions were spread throughout the semester, and therefore the time delay between Q1-Q2, quizzes and the final exam varies by question. Later questions may involve effects based on building upon concepts from earlier questions. The earliest questions may be rendered very easy by the time of the exam. Our data collapses these effects and therefore it is unclear to what extent these processes are at work.

Finally, while all the students in our sample belonged to computing majors such as CS and CE, they could have had varying industry internships leading up to our study. This could have influenced their prior knowledge in any of the six modules. We have not investigated these possible effects and intend to do so in future iterations of our study.

VII. Conclusion

In our study, we use PI questions to provide evidence that students learn from their peers and demonstrate this knowledge shortly after learning it. In this work, we replicate and extend the study by Gopal and Cooper [7] whose work uses isomorphic questions both in-class and on the final exam to address the question: do students retain and demonstrate this learning as measured by isomorphs on the final exam? We find, for each of the modules in software engineering, that students who learn from the PI process are quite likely to correctly answer questions given on not just a single exam, but module-wise quizzes as well. We find that these students are statistically as likely as students with existing prior knowledge to answer quiz and exam questions correctly. Averaged across modules, a substantial 58% of the overall class benefited from

PI. In summary, our study provides evidence that students learn software engineering concepts from in-class PI process even if held completely online, and that this learning persists weeks (or months) later on quizzes and the exam.

We believe that PI can be an effective pedagogy in teaching software engineering. The unique contribution of this study is to replicate and extend the work of Gopal and Cooper [7] in an online software engineering course comprised of computing majors. We also include two data points, several weeks apart, to establish long term cognitive retention for improved learning outcomes from PI, using fine-grained individual student response data. We plan to extend this study to multiple software engineering courses with the same methodology to establish the robustness of this approach.

REFERENCES

[1] M. D. Koretsky, B. J. Brooks, R. M. White, and A. S. Bowen, "Querying the questions: Student responses and reasoning in an active learning class," *Journal of Engineering Education*, vol. 105, no. 2, pp. 219–244, 2016.

[2] L. Porter, C. Bailey Lee, B. Simon, Q. Cutts, and D. Zingaro, "Experience report: a multi-classroom report on the value of peer instruction," in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 2011, pp. 138–142.

[3] L. Porter, D. Zingaro, and R. Lister, "Predicting student success using fine grain clicker data," in *Proceedings of the 10th Annual Conference on International Computing Education Research*, 2014, pp. 51–58.

[4] B. Simon, S. Esper, L. Porter, and Q. Cutts, "Student experience in a student-centered peer instruction classroom," in *Proceedings of the 9th Annual International ACM Conference on International Computing Education Research*, 2013, pp. 129–136.

[5] D. Zingaro and L. Porter, "Peer instruction in computing: The value of instructor intervention," *Computers & Education*, vol. 71, pp. 87–96, 2014.

[6] ——, "Tracking student learning from class to exam using isomorphic questions," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, p. 356–361.

[7] B. Gopal and S. Cooper, "Peer instruction in software engineering-findings from fine-grained clicker data," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 2021, pp. 115–121.

[8] J. Handelsman, D. Ebert-May, R. Beichner, P. Bruns, A. Chang, R. DeHaan, J. Gentile, S. Lauffer, J. Stewart, S. M. Tilghman *et al.*, "Scientific teaching," 2004.

[9] N. R. Council *et al.*, *A framework for K-12 Science Education: Practices, crosscutting concepts, and core ideas*. National Academies Press, 2012.

[10] P. L. Machemer and P. Crawford, "Student perceptions of active learning in a large cross-disciplinary classroom," *Active learning in Higher Education*, vol. 8, no. 1, pp. 9–30, 2007.

[11] J. R. MacArthur and L. L. Jones, "A review of literature reports of clickers applicable to college chemistry classrooms," *Chemistry Education Research and Practice*, vol. 9, no. 3, pp. 187–195, 2008.

[12] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "Pair programming improves student retention, confidence, and program quality," *Communications of the ACM*, vol. 49, no. 8, pp. 90–95, 2006.

[13] C. A. de Lima Salge and N. Berente, "Pair programming vs. solo programming: What do we know after 15 years of research?" in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 5398–5406.

[14] M. L. Maher, C. Latulipe, H. Lipford, and A. Rorrer, "Flipped classroom strategies for cs education," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015, pp. 218–223.

[15] K. Lockwood and R. Esselstein, "The inverted classroom and the cs curriculum," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 113–118.

[16] E. Mazur, *Peer instruction: A user's manual*. Prentice Hall, 1997.

[17] iClicker. Student Response Systems & Classroom Engagement Tools. Last Accessed July 2021. [Online]. Available: https://www.iclicker.com/

[18] C. B. Lee, S. Garcia, and L. Porter, "Can peer instruction be effective in upper-division computer science courses?" *ACM Transactions on Computing Education (TOCE)*, vol. 13, no. 3, pp. 1–22, 2013.

[19] S. N. Liao, W. G. Griswold, and L. Porter, "Impact of class size on student evaluations for traditional and peer instruction classrooms," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 2017, pp. 375–380.

[20] B. Simon, P. Kinnunen, L. Porter, and D. Zazkis, "Experience report: Cs1 for majors with media computation," in *Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education*, 2010, pp. 214–218.

[21] L. Porter, S. Garcia, J. Glick, A. Matusiewicz, and C. Taylor, "Peer instruction in computer science at small liberal arts colleges," in *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 2013, pp. 129–134.

[22] Q. Cutts, S. Esper, and B. Simon, "Computing as the 4th" r" a general education approach to computing education," in *Proceedings of the 7th International Workshop on Computing Education Research*, 2011, pp. 133–138.

[23] L. Porter and B. Simon, "Retaining nearly one-third more majors with a trio of instructional best practices in cs1," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 165–170.

[24] J. Kinne, E. Misner, A. S. Carter, and S. M. Tuttle, "Evaluating the efficacy of clicker-based peer instruction across multiple courses at a single institution," *Journal of Computing Sciences in Colleges*, vol. 34, no. 1, pp. 164–170, 2018.

[25] R. P. Pargas and D. M. Shah, "Things are clicking in computer science courses," in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, 2006, pp. 474–478.

[26] L. Porter, C. Bailey Lee, and B. Simon, "Halving fail rates using peer instruction: a study of four computer science courses," in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 177–182.

[27] N. Tan, G. Shoemaker, A. Gedi, J. Mache, and R. Weiss, "Applying a framework for creating and analyzing cybersecurity questions for peer instruction," *Journal of Computing Sciences in Colleges*, vol. 33, no. 1, pp. 102–108, 2017.

[28] T. Adawi, H. Burden, D. Olsson, and R. Mattiasson, "Characterizing software engineering students' discussions during peer instruction: Opportunities for learning and implications for teaching," *International Journal of Engineering Education*, vol. 32, no. 2, pp. 927–936, 2016.

[29] S. Esper, B. Simon, and Q. Cutts, "Exploratory homeworks: An active learning tool for textbook reading," in *Proceedings of the 9th Annual International Conference on International Computing Education Research*, 2012, pp. 105–110.

[30] S. Esper, "A discussion on adopting peer instruction in a course focused on risk management," *Journal of Computing Sciences in Colleges*, vol. 29, no. 4, pp. 175–182, 2014.

[31] Zoom. Video conferencing, web conferencing, webinars, screen sharing. Last Accessed July 2021. [Online]. Available: https://zoom.us/

[32] J. E. Caldwell, "Clickers in the large classroom: Current research and best-practice tips," *CBE—Life Sciences Education*, vol. 6, no. 1, pp. 9–20, 2007.

[33] Canvas. Canvas Learning Management System. Last Accessed July 2021. [Online]. Available: https://www.instructure.com/canvas/

[34] B. J. Ericson, L. E. Margulieux, and J. Rick, "Solving Parson's problems versus fixing and writing code," in *Proceedings of the 17th Koli Calling International Conference on Computing Education Research*, 2017, pp. 20–29.

[35] V. Razmov and R. Anderson, "Pedagogical techniques supported by the use of student devices in teaching software engineering," in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, 2006, pp. 344–348.

[36] R. Kaleta and T. Joosten, *Student response Systems: A University of Wisconsin System study of clickers (Research Bulletin, Issue 10). Boulder, CO: EDUCAUSE Center for Applied Research*, 2007.

[37] L. Porter, C. Bailey Lee, B. Simon, and D. Zingaro, "Peer instruction: do students really learn from peer discussion in computing?" in *Proceedings of the 7th International Workshop on Computing Education Research*, 2011, pp. 45–52.

[38] R. L. Wasserstein and N. A. Lazar, "The ASA Statement on p-Values: Context, Process, and Purpose," *The American Statistician*, vol. 70, no. 2, pp. 129–133, 2016.