

# Practice Makes Better: Quiz Retake Software to Increase Student Learning

Kesina Baral

Department of Computer Science, George Mason  
University, USA  
kbaral4@gmu.edu

Paul Ammann

Department of Computer Science, George Mason  
University, USA  
pammann@gmu.edu

Jeff Offutt

Department of Computer Science, George Mason  
University, USA  
offutt@gmu.edu

Rasika Mohod

Department of Computer Science, George Mason  
University, USA  
rasikamohod@gmail.com

## ABSTRACT

In the past few years, we have made several pedagogical changes to the way we teach and assess student knowledge in our courses. These courses are undergraduate software engineering courses taken in the third or fourth years, and graduate (non-research) courses taken as part of a master's degree. They are taken by software engineering majors and computer science majors. This paper focuses on a specific technique—allowing students to retake weekly quizzes. We use weekly quizzes to offer more frequent, yet lower stakes, assessments than the traditional midterm exam. Quizzes are usually given at the beginning of class meetings. We offer students who under-performed or who missed a quiz the chance to try again. A major contribution of this paper is a description of scheduling software we developed to facilitate the retake process. Retake quizzes are different from the original quizzes, but cover the same material and are of similar difficulty. Our goal is to improve student learning and retention. This paper presents a post-hoc retrospective analysis of student performance on retake quizzes. In such a scenario, only limited conclusions can be drawn. Nonetheless, we see encouraging signs that students not only achieve higher overall scores when retaking quizzes, but that some students perform better on the final exam.

## CCS CONCEPTS

• **Software and its engineering** → **Software creation and management**.

## KEYWORDS

software engineering, student learning, quiz, quiz retake, web application, software

## ACM Reference Format:

Kesina Baral, Jeff Offutt, Paul Ammann, and Rasika Mohod. 2021. Practice Makes Better: Quiz Retake Software to Increase Student Learning. In *Proceedings of the 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (EASEAI '21)*, August 23, 2021, Athens, Greece. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3472673.3473965>

## 1 INTRODUCTION

Students learn more when they receive appropriate, frequent, and detailed feedback [2]. Accordingly, one of our (Ammann and Offutt) long-standing pedagogical practices has been to give weekly, graded quizzes instead of the more traditional midterm exams. Students often comment that small weekly quizzes force them to keep up as opposed to cramming for a large exam and that they feel better prepared for final comprehensive exams after taking weekly quizzes. Of course, being forced to keep up with the learning is not always welcome to the students, but our opinion is that is positive for learning.

In our previous SPARC project [4], we separated programming practice from assessment by allowing students to submit programs for evaluation without counting them for a grade. When they felt ready, students then wrote a similar program that was assessed as part of their grade. We observed that some students' performance often improved significantly on second and third practice attempts, leading us to hypothesize that significant learning occurred as a result of the initial attempts.

A few years ago, we decided to combine these approaches by offering quiz *retakes*. The retake quizzes serve two purposes: (1) when students did poorly on the first quiz, they can study what they missed and perform better on the retake, and (2) when students are forced to miss class (earning a 0 on that quiz), they have the opportunity to earn the points on the retake. We expected (or at least hoped for) three benefits from this retake policy. By allowing students to learn from initial failures, we expected students to increase their learning. By focusing their studying on concepts or skills that they initially performed poorly on, we hoped to increase retention after the course was over. And by providing students with multiple chances to perform well on a quiz, we expected students to perform better and earn higher grades.

One of the authors (Offutt) designed support software to schedule the retakes. The immediate goal was to reduce the administrative

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*EASEAI '21, August 23, 2021, Athens, Greece*

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8624-1/21/08... \$15.00

<https://doi.org/10.1145/3472673.3473965>

load on instructors and teaching assistants for managing the retake process, but it also seemed plausible that retake software could enhance student performance by reducing friction in the retake process. This paper reports on both the (current) design and function of the retake software as well as our experience with the retake process over several course offerings. Since our analysis of these experiences is retrospective—there is no experimental design—we can only draw limited conclusions from this data.

The paper is organized as follows. Section 2 discusses our course policies for quiz retakes and section 3 presents our software design. We then briefly describe the courses that have used our retake system in section 4. Section 5 presents data from using quiz retakes, section 6 discusses related work, and section 7 provides a discussion and conclusions.

## 2 RULES AND POLICIES FOR QUIZ RETAKES

Both Ammann and Offutt have used weekly quizzes for years in many classes, and have adopted the quiz retake pedagogy in around a dozen separate course offerings. We have used this in both undergraduate and graduate software engineering courses. Courses for which we have data on the retakes are detailed in section 4.

Here, we describe our policies that coincide, then detail some slight variations that we have used. Quizzes are given in the first 10 or 15 minutes of one class meeting every week and each is worth 10 points. Some of our classes meet once a week for 2.5 hours, and some meet twice a week for 1.25 hours. Each quiz covers material from the previous week and occasionally from the reading assigned for that day, as published on the course website. Quizzes have been worth 30% of the overall course grade in Jeff’s courses, and between 40% and 50% in Paul’s courses. Graded quizzes are always returned in the next class meeting, and scores are posted online before that when instructor and TA resources permit.

The following retake policies are taken from our syllabi, and are common to both professors and all courses where we’ve used this approach. In addition to the actual policy, we also provide reasons for the policy.

- **Limitation:** *Students can have one retake per quiz.* Although in principle, we would be in favor of more than one retake, in practice that is too much work.
- **Scoring:** *The maximum score on a retake quiz is 80%.* In our judgement, offering full credit for a retake could encourage students to skip preparation for the initial quiz. We took the amount of 80% from a local high school’s policy for math exam retakes.
- **Eligibility:** *Students may only retake quizzes if the original score was less than 80%.* Also eligible are students who missed the original offering of the quiz and hence start with a score of zero.
- **Replace:** *If you retake the quiz, your new score will count and the first score is dropped.* This creates the risk of obtaining a lower score, which we hoped would encourage students to take their preparation seriously.
- **Scheduling:** *Students who want to retake a quiz must schedule ahead of time.* This helps us plan, and also helps the students to plan ahead. In practice, we usually accommodate

a few “walkups” if we had extra paper copies of the quiz. The software we used to schedule quiz retakes is described in section 3.

- **Times & locations:** *To be announced online.* We hold retake quizzes at various times and locations, including professor’s office hours, TA’s office hours, and when possible, the 30 minute slot before class meetings either in the classroom or a nearby room.
- **Content:** *The retake quiz will differ from the in-class quiz, but will cover the same topics.* We try to make the retake quizzes the same level of difficulty as the original.
- **Timing:** *The retake quiz must be taken within two weeks of the original in-class quiz. All retakes must be completed before the beginning of final exams.* The two-week window ensures that students can finish their retake while the material is still relatively fresh and ensures they get back their graded original quiz with at least a week to retake it.

Jeff and Paul’s policies with respect to retake quizzes are similar, with one exception. Although we both limit the value of the retake score to be 80% of the original (8 points), Paul designs the retake quizzes to be worth 8 points and Jeff designs them to be worth 10 points but “ceilings” scores at 8. That is, a student in Jeff’s course can miss 2 points on a quiz and still earn the maximum of 8. With the ceiling method we expect students to be more likely to increase their scores no matter what their original quiz score and to be more likely to risk a retake with a relatively high score (i.e. 6 or 7) on the original quiz.

## 3 SCHEDULING SOFTWARE

The quiz scheduling software QZSKD, is a Java J2EE web application that uses Java servlets, Java Server Pages (JSPs), Java beans, and XML to persist data. We deployed a version for public use on heroku, with source files on github<sup>1</sup>.

### 3.1 QZSKD user interface

QZSKD has two interfaces, one for students and one for instructors. Students individually create appointments for quiz retakes through the quiz scheduler interface<sup>2</sup> as shown in Figure 1. They are presented with a list of dates, locations, and which quizzes are available during the upcoming two-week window. The instructor configures this information at the beginning of the semester. Although this varies, we normally try to use instructor office hours, TA office hours, and for the convenience of part-time students who often only come to campus once or twice a week, the 30-minute window immediately preceding class meetings.

Instructors and TAs can view the upcoming schedule through the admin interface<sup>3</sup> as shown in Figure 2. This view shows upcoming retake appointments with student names.

### 3.2 QZSKD data management

Courses, appointments, quizzes, and retakes data are stored in XML files. Below is an example QZSKD XML course configuration file.

<sup>1</sup><https://github.com/Keshina/quizschedule/tree/herokuQuiz>

<sup>2</sup><https://quizscheduler.herokuapp.com/>

<sup>3</sup><https://quizscheduler.herokuapp.com/admin.jsp>

### Quiz Retake Scheduler

You can sign up for quiz retakes within the next two weeks.  
Today is *WEDNESDAY, FEBRUARY 10*. Currently scheduling quizzes for the next two weeks, until *WEDNESDAY, FEBRUARY 24*.

Enter your name (as it appears on the class roster), then select which date, time, and quiz you wish to retake from the following list.

Name:

Friday, February 12 , at 15:00 in CS Building 1000	
Quiz 2 from Friday, February 5	<input type="checkbox"/>
Quiz 3 from Friday, February 12	<input type="checkbox"/>
Saturday, February 13 , at 15:30 in CS Building 1001	
Quiz 2 from Friday, February 5	<input type="checkbox"/>
Quiz 3 from Friday, February 12	<input type="checkbox"/>
Sunday, February 14 , at 10:00 in Engineering Hall	
Quiz 2 from Friday, February 5	<input type="checkbox"/>
Quiz 3 from Friday, February 12	<input type="checkbox"/>
Friday, February 19 , at 15:00 in CS Building 1000	
Quiz 3 from Friday, February 12	<input type="checkbox"/>
Quiz 4 from Friday, February 19	<input type="checkbox"/>
Saturday, February 20 , at 15:30 in CS Building 1001	
Quiz 3 from Friday, February 12	<input type="checkbox"/>
Quiz 4 from Friday, February 19	<input type="checkbox"/>
Sunday, February 21 , at 10:00 in Engineering Hall	
Quiz 3 from Friday, February 12	<input type="checkbox"/>
Quiz 4 from Friday, February 19	<input type="checkbox"/>
<input type="button" value="Submit request"/>	

**Figure 1: Student view**

It defines parameters of each course that is currently available, including the course title, the number of days after a scheduled quiz its retake can be taken, and the location of the data files.

**QZSKD course configuration file**

```

<courses>
  <course>
    <courseID>swe437</courseID>
    <courseTitle>Software Testing & Maintenance</courseTitle>
    <retakeDuration>14</retakeDuration>
    <dataLocation>/data/</dataLocation>
  </course>
  ...
</courses>
```

The Quizzes XML file, shown next, defines each quiz in the course. The quiz **id** must be a unique positive integer, and the month, day, hour, and minutes usually correspond to class meetings. Values are checked for conformance to a valid date and time.

**Quizzes XML file**

```

<quizzes>
  <quiz>
    <id>1</id>
    <dateGiven>
      <month>2</month>
      <day>5</day>
      <hour>10</hour>
      <minute>30</minute>
    </dateGiven>
  </quiz>
  ...
</quizzes>
```

The Retakes XML file, shown next, has a similar format, but also includes a location for the retake. Locations vary depending on who administers the retake and when.

```

Retakes XML file
<retakes>
  <retake>
    <id>1</id>
    <location>Engineering 404</location>
    <dateGiven>
      <month>9</month>
      <day>4</day>
      <hour>15</hour>
      <minute>30</minute>
    </dateGiven>
  </retake>
</retakes>

```

The current version of the QZSKD system requires instructors to build XML files by hand at the beginning of the semester. This is also not a commercial product, so we do not provide a deployment platform. Heroku is free and integrated with github, but limited. The software also provides the ability to skip holidays such as spring breaks.

### 3.3 QZSKD software design

QZSKD implements the Model-View-Controller (MVC) design pattern [5] to separate the application control (the *Controller* layer), the two user views (the *View* layer), and the data representation and application logic (the *Model* layer). The Controller layer is implemented with Java servlets, the View layers are implemented with JSPs, and the Model layer is implemented with Java classes, including classes that follow the JavaBean convention. The view has four JSPs to handle request pages and response pages for students and instructors. The model consists of application logic related to appointments, retakes, quizzes, and courses. The model is also connected to the application's data structure, stored as XML files.

Figure 3 shows details of our system. The View layer has two views; `appResponse.jsp` and `index.jsp` are for the students to schedule retakes, and `quizScheduleForm.jsp` and `admin.jsp` are for instructors and TAs. The Controller is a single servlet and the Model has several components; an Appointment object, a Retake object, a Course object, and a Quiz object.

## 4 COURSES THAT USED RETAKES

We have been using the quiz retake system since the fall semester of 2017, in both graduate and undergraduate software engineering courses. The next section in our paper presents results from eight course sections across three different courses. In fall 2017 and fall 2018, Jeff gave retake quizzes in the Master's level course SWE<sup>4</sup> 637, Software Testing. The 2017 course had 36 students and the 2018 course had 40. The 2018 syllabus can be viewed on the web: <https://cs.gmu.edu/~offutt/classes/637/>. Quiz retakes were given in SWE 437, Software Testing & Maintenance, in spring 2018 (59 students), in spring 2019 (60 students), and in fall 2019 (60 students). The fall 2019 syllabus can be viewed on the web: <https://cs.gmu.edu/~offutt/classes/437-2019fall.html>.

In fall 2018, Paul gave retake quizzes in SWE 332, Object-Oriented Software Design and Implementation. The course had

30 students. The most recent syllabus can be viewed on the web: <https://cs.gmu.edu/~pammann/332.html>. In fall 2018, Paul gave retake quizzes in SWE 437, Software Testing & Maintenance. The course had 40 students. The fall 2018 syllabus can be viewed on the web: <https://cs.gmu.edu/~pammann/437.html>. In spring 2019, Paul gave retake quizzes in SWE 637, Software Testing. The course had 25 students. The spring 2019 syllabus can be viewed on the web: <https://cs.gmu.edu/~pammann/637.html>.

Both teachers were using retakes in spring 2020, but when we made an emergency mid-semester switch to online teaching due to the pandemic, we stopped giving retakes. Fall 2020 and spring 2021 were also online without retakes, again due to the pandemic. We plan to return to using retakes when classes resume in-person in fall 2021.

## 5 EFFECTS AND OUTCOMES

This section presents quantitative data and qualitative observations from the use of quiz retakes across the eight course offerings identified in section 4. We rely on data (quiz and exam scores) collected for educational purposes. This is a post-hoc data analysis, that is, we did not gather data according to an experimental design<sup>5</sup>. We anonymized all grades and data before performing analysis and only present aggregate, non-identifiable, measures.

### 5.1 Quantitative results

Table 1 presents descriptive data from our courses. The courses are ordered by course number and by semester offered, for example, "18F" means Fall semester 2018. For each course, the table gives the number of quizzes (with a few exceptions, each week of class meetings except the first week) and the number of students who completed the course. *Complete* means the student took the final exam and was assigned a grade, and thus includes students who failed the course, but not students who dropped during the semester. The *weight of quizzes* is relative to the overall course average. The *# of 0 scores not retaken* is the number of times a student did not take the initial quiz, and then did not take advantage of the retake (or "late-take").

Course policies allow students to retake quizzes when their initial scores are less than 8. Since another policy allows scores to decrease, in practice, when the initial scores is 6 or 7, the risk of a lower score can out-weigh the potential 1 or 2 point benefit of retaking the quiz. Since less than 0.7% of all retakes were from initial scores above 6, in Table 1 we consider retakes of quizzes with initial scores of less than 6, so the *% eligible quizzes retaken* is the number of quizzes retaken divided by the eligible quizzes.

We define *quiz retakers* to include students who retook at least one quiz, excluding students who did not retake any quizzes. Since students who did not retake quizzes could not benefit from the policy, this allows us to measure the benefit of retaking quizzes among the students who tried. Thus, the *Average increase of retakers* is the average increase in the *retaker's* overall semester quiz average. For example, if the course had 10 quizzes of 10 points apiece, and a student retook two quizzes, changing one score from 4 to 8, and the other from 6 to 8, the average increase for that retaker would be .6.

<sup>4</sup>George Mason abbreviates Software Engineering as "SWE," because "SE" was already used by Systems Engineering.

<sup>5</sup>For this same reason, we were assured by our IRB office that we were in compliance with IRB requirements without a full formal IRB review.

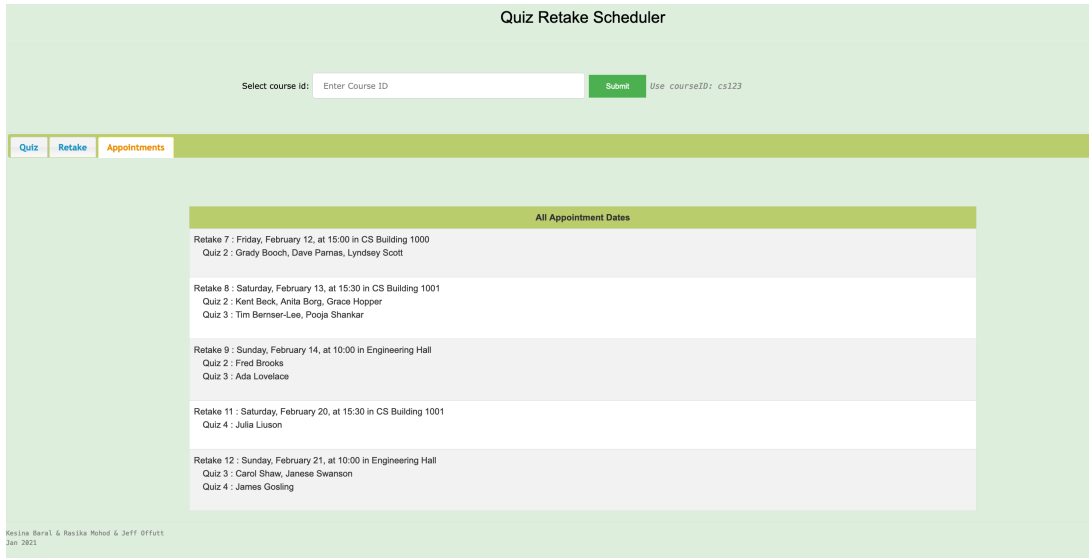


Figure 2: Instructor and TA view

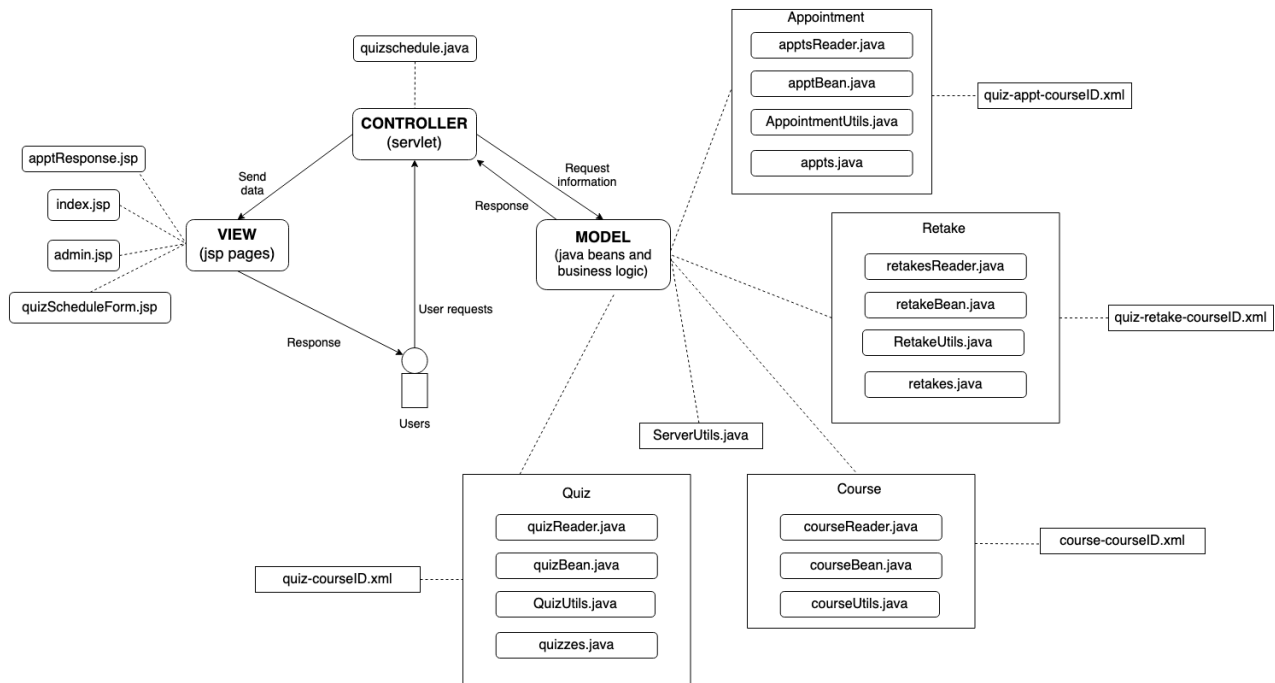


Figure 3: QZSKD software architecture

All quizzes in all courses were worth 10 points apiece. When quizzes were worth 30% of the overall grade, an average increase of .5 in the quiz score would equate to an average increase of 1.5 in the overall point score.

Next we performed a post-hoc analysis of the effect of retaking quizzes on final exam scores. We observed that some students had limited understanding of concepts and difficulty applying those

concepts when taking initial quizzes. By struggling in their first attempt, and not performing well, they realized what they were missing. Then they re-studied the material, and often did better on the retake quiz. In other words, their first attempt was practice, and the process of practice, additional studying, and retakes allowed them to perform better.

**Table 1: Summary descriptive data from eight course offerings that used quiz retakes, including two instructors, three courses, and five semesters**

Measure	Semester Instructor	332 18F	437 18S	437 18F	437 19S	437 19F	637 17F	637 18F	637 19S
		Paul	Jeff	Paul	Jeff	Jeff	Jeff	Jeff	Paul
# quizzes		13	12	12	13	13	13	13	13
# students		30	59	40	60	60	35	40	25
% weight of quizzes		40.0%	30.0%	45.0%	30.0%	30.0%	30.0%	30.0%	50.0%
# of 0 scores not retaken		3	6	15	20	16	2	1	1
% all quizzes retaken		11.3%	20.3%	8.8%	17.4%	18.1%	17.8%	11.9%	6.2%
% eligible quizzes retaken		45.0%	89.4%	38.0%	81.0%	72.3%	76.0%	100.0%	28.0%
% students retook at least one		66.7%	98.3%	55.0%	85.0%	71.7%	85.7%	75.0%	44.0%
Average increase of retakers		0.49	0.85	0.50	0.76	0.89	1.18	0.61	0.67

**Table 2: Correlations between quiz average and final exam scores with and without retakes**

Course	Semester	Without retakes	With retakes	Diff
SWE 332	18F (P)	-0.80	-0.73	+0.07
SWE 437	18S (J)	0.58	0.76	+0.18
	18F (P)	-0.07	0.38	+0.45
	19S (J)	0.01	0.09	+0.08
	19F (J)	0.30	0.73	+0.43
SWE 637	17F (J)	0.32	0.38	+0.06
	18F (J)	-0.09	0.50	+0.59
	19S (P)	0.69	0.94	+0.25

We hypothesised that this process also increased their learning, which could result in better performance on the final exam. We isolated this effect in the following way. When students missed the initial quiz offering, and then retook the quiz, that was an entirely different process (“late-takes” as opposed to “retakes”). So we started by eliminating those data.

Table 2 presents results across our eight courses. Since it seems likely that the effect on final exam score would be greater for students who lost a lot of points on initial takes of quizzes, we considered only students who scored in the lowest quartile on the initial takes. We computed Pearson’s correlation between the average scores students would have gotten if they did **not** take any retakes (*Without retakes*) with their final exam scores, and the same correlation using the average scores students got **after** taking the retakes (*With retakes*). The *Diff* column is the difference in the two correlations.

## 5.2 Discussion and observations

It is important to note that our observations cannot be conclusive. First, this is a post-hoc analysis of existing data, so we did not start with a hypothesis and could not apply experimental controls. Second, we have a relatively small sample of data—a total of about 350 students. Further, the correlations in table 2 are on quartiles over students eligible for at least one retake *at the course level* and hence include only between 6 and 15 observations, depending on the class. Third, we have data from three different courses at the undergraduate and graduate level, so there was a lot of variation in how quizzes were handled.

Table 1 yields several interesting observations. From the “% eligible quizzes retaken,” we note that a lot of students simply do not take the opportunity. In fact, many students who did not take the initial quiz did not retake a quiz—effectively accepting the zero. When comparing the graduate courses (SWE 637) with the undergraduate courses (SWE 332 and SWE 437), we note that graduate students are more likely to retake quizzes. In particular, almost every graduate student who missed the initial quiz retook it.

We also note that a larger percentage of students in Jeff’s courses retook quizzes than in Paul’s courses, even though Paul’s courses weighed quiz averages higher. One plausible hypothesis is that the “ceiling” approach to grading retake quizzes in Jeff’s courses encouraged more students to retake quizzes.

Table 2, although far from definitive, is at least encouraging. Recall that Pearson’s correlation coefficient is measured from -1 to +1, where 0 indicates the two groups of numbers had no correlation at all, +1 is an exact positive correlation, and -1 is an exact negative correlation.

Although the actual correlations between quiz average after retakes and final exam varied, the correlations with retakes was consistently higher for all eight courses. Although some courses had a very small increase (+.04), several were quite high (.36, .43, and .59). SWE 332 is somewhat of an outlier, in that we found a strong negative correlation between quiz scores and final exam scores. Our only explanation for these numbers is that the dataset is small.

Overall, despite the small numbers of scores available, we believe this analysis gives some support for the theory that retaking quizzes helps students learn more, especially struggling students who need more help. Although we are tempted to note that students who struggle in software engineering courses often include a high percentage of first-generation students, female students, and under-represented minorities, we do not believe this study has enough data to conclusively show that allowing quiz retakes helps make our courses more inclusive.

## 6 RELATED WORK

Tanaka et al. [6] analyzed the effect of using gamified quizzes on student motivation. The study shows that 44% of the student reported high motivation for quiz preparation when game elements like badges, progress levels, and leader-boards were used. Crawford [1] performed a study where both students’ lecture notes and

quizzes were graded and compared to determine if there is any correlation between the two. The study reports that notes that were complete, clear, and correct were closely related to higher quiz scores than notes that were brief, unclear, and incorrect. Lalopa et al. [3] conducted a study to analyze if creating student teams with performance-based incentive packages affected their quiz scores. They report that student teams got consistently high quiz scores with the performance-based incentive packages. Similarly, the possibility of improving quiz scores and getting better grades acted as an incentive for students to retake quizzes that did not go well the first time. Hattie and Timperley [2] performed a much more general study on the effect of feedback on learning. In our context, quiz retakes most closely map to providing feedback about processing of tasks (FP) in Hattie and Timperley's four level feedback scheme.

## 7 DISCUSSION AND CONCLUSIONS

We hypothesize that the QZSKD software can help reduce the cost of offering retakes, but with two caveats. First, since the analysis is post-hoc, we do not have data on the time spent offering retakes either with or without the QZSKD software. Second, there are other costs to offering retakes: (1) additional grading, (2) time designing additional quiz questions (for example, designing one 20-point quiz then splitting it in half), and (3) staffing the retake quiz offerings. These costs may limit scalability, especially if there is a shortage of graduate or undergraduate TAs.

There is also a potential integrity loss due to the same quiz retake being given to different students at different times. We limit this by not returning graded retakes while students are still eligible for a retake on that particular quiz. While this mechanism is imperfect, we believe that the potential for increased learning outweighs the potential for integrity loss.

It is worth noting that many students have expressed strong appreciation for the retakes, both in anonymous comments on end-of-semester student evaluations and from conversations with students in our classes.

Several patterns repeated enough to capture our attention. One pattern is that some students, mostly undergraduates, regularly bombed the first quiz, then aced the second. We believe these students used the initial attempt to study for that particular quiz and were, somewhat to our surprise, prepared to pay a 20% penalty for the privilege. Another pattern is that some students simply ignore the opportunity for retakes (or late-takes) and accept the original grade, even if it is a zero. We observed that this happens mostly with undergraduates and is rare among graduate students.

We emphasize that due to the post-hoc nature of our data analysis, Table 2 should be viewed mostly as an encouraging invitation for further study rather than a definitive result. That is, overall, students' final exam grades tend to track their retake quiz averages more

closely than their original quiz scores, which suggests that learning took place during the retake process.

The QZSKD software is available for anyone to use as is or extend as desired. We believe the MVC separation of concerns makes the design easy to modify. That said, the software is not a finished product ready to be used by instructors unfamiliar with the technologies used to implement the software. One extension we would like to add is an interface to create the schedules, thereby avoiding the need to edit XML directly. Another obvious extension is an implementation of a general authentication mechanism. We currently use https authentication to protect the retake scheduling data, but a more robust solution that does not require access to file permissions on the web server would be better.

It has been suggested that offering retakes for quizzes make the course "too easy," particularly when the quiz average is a high percentage of the grade. We emphatically reject this hypothesis philosophically and pedagogically. As teachers, we view our responsibility is to make learning as easy as possible. The material is complex ("hard"), abstract ("hard"), and most students need substantial time to learn how to apply the concepts in practice (more "hard"). Anything we can do, as teachers, to reduce the challenge of learning this material helps our students, their eventual employers, and people who use their software. Note that we are **not** saying that grades should be easy, rather, **learning** should be easier. We believe that our system increases learning while reducing the pain of learning.

The data and our experience with quiz retakes lead us to conclude that the technique helps students who are struggling with the material the most, which is exactly who we want to help. While we would certainly like more definitive results, we are generally satisfied with the value of the retake technique and intend to continue using it.

## REFERENCES

- [1] C. C. Crawford. 1925. The Correlation between College Lecture Notes and Quiz Papers. *The Journal of Educational Research* 12, 4 (1925), 282–291. <https://doi.org/10.1080/00220671.1925.10879600>
- [2] John Hattie and Helen Timperley. 2007. The Power of Feedback. *Review of Educational Research* 77, 1 (March 2007), 81–112.
- [3] Joseph M. Lalopa, James W. Jacobs Jr., and Cary Countryman. 1999. The Effects of Student Teams on Quiz Score Performance Given a Performance-Based Incentive Package. *Journal of Hospitality & Tourism Education* 11, 2-3 (1999), 25–30. <https://doi.org/10.1080/10963758.1999.10685231>
- [4] Jeff Offutt, Paul Ammann, Kinga Dobolyi, Chris Kauffman, Jaime Lester, Upsorn Praphamontipong, Huzefa Rangwala, Sanjeev Setia, Pearl Wang, and Liz White. 2017. A Novel Self-Paced Model for Teaching CS1 and CS2. In *Learning at Scale*. ACM Press, Boston, USA, 1–4.
- [5] Trygve Reenskaug and James O. Coplien. 2009. The DCI Architecture: A New Vision of Object-Oriented Programming. Online. <https://www.artima.com/articles/the-dci-architecture-a-new-vision-of-object-oriented-programming>, last access May 2021.
- [6] Yuki Tanaka, Hidetake Uwano, Tomohiro Ichinose, and Shinya Takehara. 2016. Effects of Gamified Quiz to Student's Motivation and Score. In *International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*. IEEE Computer Society Press, Barcelona, Spain, 1–4.