Student Perceptions of Fairness and Security in Versioned Programming Exams

Chinedu Emeka
Computer Science
University of Illinois at Urbana-Champaign, USA
cemeka2@illinois.edu

Craig Zilles
Computer Science
University of Illinois at Urbana-Champaign, USA
zilles@illinois.edu

ABSTRACT

Using multiple versions of exams is a common exam security technique to prevent cheating in a variety of contexts. While psychometric techniques are routinely used by large high-stakes testing companies to ensure equivalence between exam versions, such approaches are generally cost and effort prohibitive for individual classrooms. As such, exam versions practically present a tension between exam security (which is enhanced by the versioning) and fairness (which results from difficulty variation between versions).

In this work, we surveyed students on their perceptions of this trade-off between exam security and fairness on a versioned programming exam and found that significant populations value each aspect over the other. Furthermore, we found that students' expression of concerns about unfairness was not correlated to whether they had received harder versions of the course's most recent exam, but was correlated to lower overall course performance.

KEYWORDS

fairness, exam security, assessment, programming, randomized exams, question variants

ACM Reference Format:
Chinedu Emeka and Craig Zilles. 2020. Student Perceptions of Fairness and Security in Versioned Programming Exams. In *Proceedings of the 2020 International Computing Education Research Conference (ICER '20), August 10–12, 2020, Virtual Event, New Zealand.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3372782.3406275

1 INTRODUCTION

In the U.S. and internationally, computer science departments are observing enrollment surges resulting from both increased numbers of majors and more courses taken by non-majors [4, 15, 20, 33]. In courses with hundreds or even thousands of students [10], one of the key challenges is performing assessment, because exam logistics (e.g., proctoring, conflict exams) scale poorly with enrollment. In addition, because performance in early courses in the CS curriculum can significantly influence who gets admitted to the major [28], exam security in these early courses may be particularly important.

One commonly used exam security technique is question variants [19, 37], which can be employed in a number of different ways. In large courses, instructors may have multiple versions of a paper exam to prevent cheating between students who are seated in close proximity to one another. Many learning management systems enable generating individualized assessments by drawing questions randomly from a question bank. Instructors may also ask different questions on a conflict exam than on a main exam for a course. Faculty try to design these variants such that they have equal difficulty, but the process is imperfect. Accounts from the standardized testing industry indicate that the cost of rigorous item generation is very high; for instance, a cost of $1,500 to $2,500 per question has been reported for the GMAT [31]. As such, the use of question variants in conventional classrooms generally creates an inherent tension for faculty between the desire to ensure exam security (through variants) and the practicality of guaranteeing fairness.

We hypothesized that a similar tension exists for students. From previous work (discussed in Section 2), we know that most students do not want to cheat and, therefore, want faculty to create courses where cheating is not a productive strategy. Previous work also shows that students become resistant to instruction and learning is hampered if students feel an instructor is unfair. There is a gap in the research literature, however, in understanding how students perceive the relationship between exam security and fairness resulting from question variants.

We believe that understanding student perceptions of this trade-off is salient for instruction in computer science—in early programming courses in particular—for two reasons. First, programming courses benefit significantly from running assessments on computers. Computer-based assessment of programming skills is both more ecologically valid (because compilers and debuggers can be used) and more efficient (because the student code can be autograded[1]) than pencil-and-paper exams. As discussed in Section 2, a common approach for computer-based assessment in early programming courses is by running "lab exams" where students take the exams in their existing lab sections with different sections receiving different question versions. Second, programming exams often contain a few large programming questions that each are worth a significant portion of the exam's points. When a question is worth a lot of points, difficulty variance between versions potentially translates into significant unfairness.

As such, this paper asks the following research questions:

ICER '20, August 10–12, 2020, Virtual Event, New Zealand
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7092-9/20/08…$15.00
https://doi.org/10.1145/3372782.3406275

[1] While manual grading for style and the assignment of partial credit is valuable, needing humans to verify a functionally correct submission by compiling hand-written student code in their heads is incredibly inefficient.

RQ1 What are computing students' perceptions of the practical trade-offs between exam security and fairness posed by question variants?

RQ2 What factors contribute to the variations in student perceptions?

We investigated these questions through a mixed methods analysis of a survey that included open-ended questions posed to a sophomore-level data structures course that used question variants. The course and the survey are described in Section 3. In Section 4, we describe the qualitative methods used to analyze the survey results and present our findings related to RQ1. Notably, we find that some students seem to be more concerned with exam security while other students seem to be more concerned with fairness. This finding naturally led to RQ2. In exploring RQ2, we posed the following two hypotheses:

H1 Students that express concerns about unfairness are more likely to have received harder question versions on the most recent exam.

H2 Students that express concerns about unfairness are more likely to be lower performing students in the class as a whole.

In Section 5, we perform a statistical analysis to explore how students' expression of "security-focus" or "unfairness-focus" relates to which questions they received on the exam prior to the survey and their overall course performance. We find no support for H1, but H2 is supported to a statistically significant degree. We discuss the implications of these findings in Section 6. We present limitations in Section 7 and conclude in Section 8.

2 RELATED WORK

2.1 Student perceptions of exam security

Studies routinely find that a significant fraction of students admit to having cheated [23, 27, 39, 40]. In spite of this, students have stronger attitudes against academic dishonesty than faculty typically perceive them to have [21]. For example, one survey found that 84% of students agree with the statement "under no circumstance is cheating justified" [2], which is surprising given that in most surveys significantly more than 16% of students admit to having cheated.

How do we make sense of these conflicting statements by students? One plausible explanation is that the majority of students don't want to cheat but some struggle to resist in an environment that rewards cheating. McCabe found students often rationalized cheating based on the unethical actions of their peers. Students may contend that "they have no choice when a faculty member makes little or no effort to prevent or respond to cheating" [27]. Steininger found that a number of factors including "professor leaves [during an exam]" and "professor discovers cheating rarely" lead students to rate cheating as more justified in hypothetical exam scenarios. One common student cheating justification is the perception that everyone else is doing it [14].

Many students believe it is the faculty's job to prevent cheating. A survey of students at two universities without honor codes found that the concept of an honor system was not popular (desired by only 28% of surveyed students); a plurality of students preferred a "system in which the faculty do the police work, while students serve as trial judges" [25]. Question randomization was perceived by students as the most effective deterrent to cheating in online exams [37].

2.2 Framework for examining fairness

Instructors' practices are the basis for students' perceptions of fairness in courses [22, 30]. Perceived fairness throughout the duration of a course, particularly with respect to grading, is a substantive determinant of students' motivation, learning, and attitudes towards an instructor [13].

Organizational justice has been previously used in the educational setting as a framework for reasoning about fairness [11, 12, 29]. Organizational justice refers to theories of interpersonal fairness which help develop understanding of organizational behavior. Organizational justice is comprised of distributive, procedural and interactional justice [22, 30], though interactional justice is sometimes combined with procedural justice in the literature [13].

Distributive justice relates to perceptions of the fairness of outcomes based on the result of an allocation decision [13]. Distributive justice is based on comparisons; individuals evaluate their relationships based on inputs they make and benefits they receive [1, 36]. When a student receives a grade, the student will often assess the amount of credit awarded and determine if it is fair based on comparisons to some standard, such as the student's expectations or the performance of their peers with similar work. Even a good grade may seem unfair in comparison if a peer achieves a better score for lower quality work. This is related to equity theory, which states that exchanges are deemed fair if individuals receive outcomes they believe they deserve based on their contributions [1]. Over-rewarding may lead to guilt, and under-rewarding may lead to anger. Previous work has shown a positive correlation between outcomes fairness and student learning [26]. Students who got grades that were consistent with their expectations given their inputs were satisfied.

Procedural justice refers to perceptions of the fairness of procedures used in making allocation decisions; procedural justice is different from distributive justice in that distributive justice centers around the fairness of the allocation decision itself, while procedural justice is focused on processes used to reach that decision. Techniques used may include meritocratic and particularistic grading practices [22]. In a meritocratic system, students receive grades based on their performance on certain components of a course (i.e., academic achievement). In a particularistic grading system, a student is evaluated on the basis of individual characteristics or circumstances [22]. For instance, a grade school student transferring from another district may not have the same pre-requisite knowledge as the rest of their class and so the student may be graded using a different standard. When the process used to compute students' grades differs, it may be considered unfair.

The third component of organizational justice is interactional justice, which deals with the social sensitivity of teachers when they address grading issues. It encompasses transparency in grading and a willingness to discuss why students received the grades they did [22]. For example, if an instructor takes time to explain why a student received a B on an essay and provides specific comments about weaknesses or areas for improvement, the student may perceive interactional justice.

Combined, these three components of organizational justice form the basis for understanding students' perceptions of just or unjust treatment in the classroom.

Our work is most related to procedural justice because the use of question variants means that the process of computing two students' grades is not identical. Students that perceive that they received more difficult questions will feel that their desire for procedural justice has been violated and deem the situation to be unfair.

2.3 Previous work of students' perceptions of fairness of versioned exams

The mostly closely related work to this paper is a study involving computer science exams where students received different versions of multiple choice questions (MCQ) [17]. On these exams, all students received the same questions, but the choices of available answers varied among students in an effort to prevent students from trivially sharing correct answers with their classmates. The questions were designed in such a way that multiple correct answers were possible, so exam questions were constructed by selecting a correct answer and N-1 incorrect answers from a pool of correct and incorrect answers, respectively. The researchers surveyed students about their perceptions of the fairness of this approach. Their findings have significant overlap with ours. They found that students strongly agreed that using question variants helped to reduce cheating. Furthermore, they found that some students have fairness concerns about using question variants, while other students are satisfied with the "personalized" exam approach and believe that if students understand the material well then variants are not a problem. Our study goes one step further and relates these student opinions to the specific questions the students received and their overall course performance in an effort to understand the source of these different perspectives.

A second study investigated students' perceptions about a wide range of aspects of online assessments ("e-assessments"), including contribution to learning, practicality, security and reliability of such tests [18]. Included in the survey was a single 5-point Likert scale item relating to question variants:

> Randomized questions from a bank means that sometimes you get easier questions.

Students slightly agreed with this statement (3.34 where 3.0 would be "neither agree or disagree"), but it represented the greatest perceived concern from students about e-assessments. This single Likert scale item gives little insights about how students think about this unfairness nor about how the unfairness inter-relates with the exam security motivation of item banking.

2.4 Computer-based Assessments

There is a long history of using computer-based exams in introductory programming courses. An early example is the 5-hour proctored end of semester "mastery" exams at Carnegie Mellon University (CMU) administered in 1985 where students were given one of a collection of similar problems to solve on a computer [5]. To accommodate a class larger than the available computer lab, students were given a choice of exam slots over a two week period. More recently, these exams have been referred to as "lab exams" because they are offered during lab sections.

Lab exams have been shown to be effective. After introducing lab exams, Califf *et al.* saw a drop of the withdrawal/D/F rate in a follow-on programming course from 28.9% to 18.2% [3]. Chamillard observed that lab exam scores correlated better with written tests and the final exam than traditional programming assignments did [6].

Most relevant to this paper, lab-exam classes typically produce multiple versions of the exam to prevent cheating from both reading off a neighbor's screen and due to the asynchronous nature of the exams [3, 7, 24, 32, 34]. Previous work provides evidence that question versions mitigate cheating on asynchronous exams [9]. These multiple versions created for security potentially present a fairness concern.

3 DATA

We conducted this study in a sophomore-level data structures course at a large public U.S. university in the Spring 2019 semester. This course is required for computer science (CS) majors, CS minors, and computer engineering majors. The vast majority of the 453 students enrolled were traditional college age students (346 male, 107 female) and 36% of the students were international.

The data that we analyzed was drawn from a course survey administered about a week after the 4th mid-term exam (out of 6) of the semester. The exam was administered during the 10th week of the 15 week semester. The exam consisted of two programming questions and students had 1 hour and 50 minutes to complete the exam. We believe that time pressure was not students' primary obstacle, as only 11% of the exam points were earned in the final 20% of the exam time.

The exam was conducted in a Computer-Based Testing Facility (CBTF) [41, 42], which permitted students to compile, test, and debug their code on their respective local computers before submitting it for grading. The networking and file systems of this computer lab are controlled to prevent students from communicating or accessing unallowed materials [43]. Because the course enrollment is much larger than the CBTF and for the convenience of the students, the exam is run asynchronously with students selecting their preferred exam time from a three day window.

The exam was autograded for function; no points were awarded for coding style or formatting. Grading was based on a collection of test cases.[2] Students had an unlimited number of attempts, and there was no penalty for successive attempts.

3.1 Question Pools

In an effort to mitigate collaborative cheating (where early exam takers inform later exam takers what is on the exam), exams in the course are randomized. This 4th mid-term exam consisted of two pools of questions, and students were given a random draw from each pool. Each pool is designed to address a particular learning objective and have a target difficulty. The first pool was intended

[2] There was some variation in the number and weighting of test cases for the Q1 variants (see Section 3.1), because the more significant question differences led to difference in testing structure. The parallel nature of the Q2 variants meant each version had equivalent test cases.

Table 1: Question 1 Variants

Question
Implement a queue using a `stl::vector`
Implement a stack using a `stl::vector`
Recursively clone a binary tree
Implement a find function for binary search tree
Implement insertion for a binary search tree
Recursively create a mirror image of a binary tree

Table 2: Question 2 Variants

Question
Pre-order traversal of only even tree elements
Pre-order traversal of only negative tree elements
In-order traversal of only nonnull tree elements
In-order traversal of only odd tree elements
In-order traversal of only positive tree elements
Pre-order traversal of only positive tree elements

Table 3: Pre-exam Information on Programming Questions

Question 1: one of
Implement stack or queue using stl::vector
Make a copy of tree or its mirror image
Implement find or insert on a BST
Question 2: implement an iterator on a binary tree
Pre-order or in-order
From left-to-right or right-to-left
Subset of elements (e.g. odd values, non-nulls, negative)

to be an easier set of questions relating to impementation of stacks, queues, and (non-balancing) binary trees; the second pool was focused on writing C++ iterators to traverse binary trees, which was expected to be harder. The questions in each pool are shown in Table 1 and 2. Students had written versions of all of these problems previously on labs or programming assignments, but the base code for the exam versions was changed so that students could not simply memorize solutions from their homework.

Because writing an iterator for a binary tree is a challenging task in the context of a closed-book timed exam, the instructor was concerned that students who had heard that iterators were on the exam would have a significant advantage preparing for the exam over students that were less connected. To mitigate the potential advantage from collaborative cheating, the course staff decided to provide students with not only a description of the structure of the exam but also high-level descriptions of the questions in both pools. The information that was provided is shown in Table 3, which in this paper is referred to as the *pre-exam* information.

For most pools of programming questions in the course, the average scores of the problems within a given pool only varied around 10%. In pool 1 of Exam 4, however, the gap between two question versions in the Q1 pool, `mirror` and `stack`, was 26%. In Figures 1 and 2, we plot the mean points earned per question variant with 95% confidence intervals (CI) for the Q1 and Q2 pools, respectively.

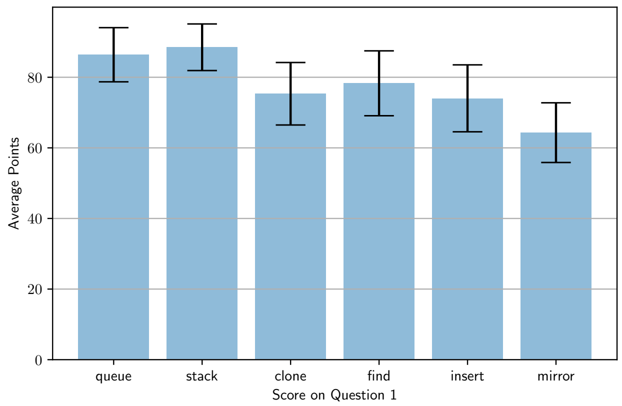

Figure 1: Average Points for Question 1 Variants

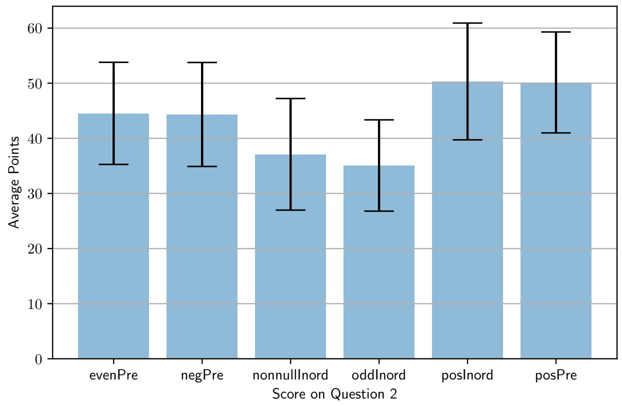

Figure 2: Average Points per Question 2 Variants

Using one-way ANOVA to compare the mean scores for each variant, we find that variants within the Q1 pool are statistically significantly different in difficulty; the F statistic was 4.437 with df = (5, 447) and p = 0.0006. Post-hoc two-sample t tests indicated that `mirror` differed from both `stack` and `queue`. For `mirror` compared to `stack`, the test statistic t = -4.399, with df = 165 and *p-value* < 0.00001. For `mirror` compared to `queue`, the test statistic t = -3.665, with df = 147 and *p-value* = 0.0004. For Q2 variants, the F statistic was 1.749 with df = (5, 404) and p = 0.1225, indicating that there was no statistically significant difference between the students' performance on the versions. To verify that the perceived difference in difficulty in the Q1 pool was a function of the question itself and not the subpopulation that received that question, we compared the performance in the rest of the course for each question's subpopulation and found no statistically significant differences in performance.

Because students had been expressing perceptions of unfairness in course forums, the instructor conducted a survey to decide how to resolve this issue.

3.2 Survey

About a week after all students had completed the exam, they were asked to fill out a survey online. The survey asked about their experience with the exam in question. Relevant to the present study

Table 4: Relevant Survey Questions

Question Dimension	Question
Exam Security	Because our exams are asynchronous (different people can take them at different times), we have multiple versions of every question so that one person can't trivially tell other people the answers to the exam questions. How important to you is it that we take steps to prevent cheating?
Exam Difficulty Variance	While we do our best to ensure that the exams are of very similar difficulty, it is impossible to make all exams have identical difficulty. In light of your ... desire for exam security, what would you like to share with us related to variations in exam difficulty?
Pre-Exam Utility	Exams necessarily only cover a fraction of the material covered in a course, and one source of variance in student exam scores results from how closely the material studied by the student matches the material on the exam. We sought to reduce this variance by being transparent about what the exam questions would be. How valuable did you find the pre-exam descriptions of the exam questions?

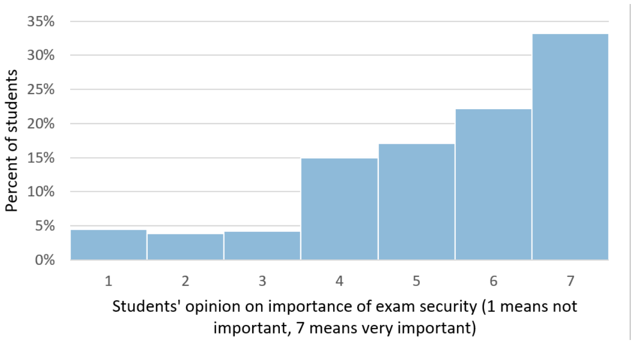

Figure 3: Students desire instructor to prevent cheating. Student rating for: "How important to you is it that we take steps to prevent cheating?"

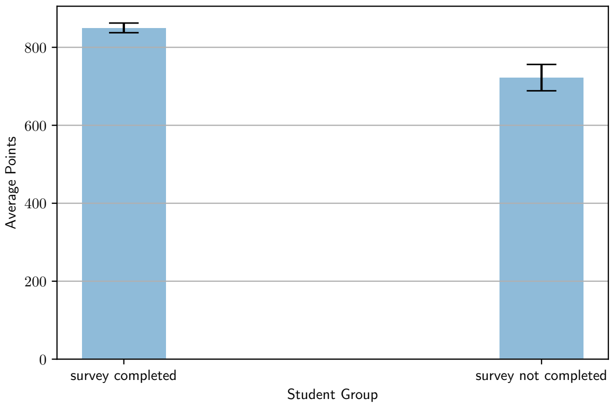

Figure 4: Survey respondents out-performed non-respondents in the course. Average overall points (out of 1000) by survey completion.

are the three prompts shown in Table 4. The survey was not anonymous; a small amount of extra credit was provided as an incentive to complete the survey and students needed to authenticate so that the instructor knew which students to grant the extra credit.

Each of these prompts provided an opportunity for an open-ended written response. It is the analysis of those open-ended responses that are the primary focus of this paper. The first prompt also included a 7-point Likert scale item. The vast majority of students indicated that the instructor taking steps to prevent cheating was important to them, with almost a third of respondents ranking it at the highest level, as shown in Figure 3.

The survey was completed by 335 of the 453 students, a response rate of 74%. The instructor matched each survey response to two other data records: (1) the versions of the questions and the scores received by the student on the Exam 4, and (2) the student's overall point total in the course. These records and the corresponding information for non-respondents were then anonymized and provided to the researchers. One survey response was discarded because it was completed by a student that had not taken Exam 4.

Notably, the survey respondents appear not to be a random sample of the course as a whole. Survey respondents have statistically significantly higher semester point totals than non-respondents (Figure 4). The 95% confidence interval for the difference between the means was (91.5, 163.7) out of a possible 1000 points (effect size: Cohen's $d = 1.108$). The test statistic t for a two sample t-test was -8.65, $df = 458$ and $p\text{-}value < 0.00001$. The amount of extra credit provided to those who completed the survey is too small to account for this difference in performance between respondents and non-respondents. We suspect that non-responders have lower motivation or organization skills that also impact their course performance.

4 QUALITATIVE ANALYSIS

We used grounded theory to analyze the written responses from the student survey. Grounded theory is an inductive reasoning research method that can be employed to develop theories from a body of data without an *a priori* hypothesis [8, 35].

In this study, two researchers coded all of the data. For each of the three questions, the researchers coded independently, developing their own set of descriptive tags. The researchers then met in successive rounds (for each survey question) and reconciled the code book and the set of tags for each student comment. The number of codes assigned to each student comment varied, with some off-topic responses receiving no codes and other comments receiving several codes.

The reconciled code book included 81 codes. We calculated the percentage of agreement by computing the total number of instances of agreement, then dividing that by the total number of instances of agreement and disagreement. The value was found to be 0.740.

Table 5: The fifteen most common codes across the three questions. Because student comments have more than one code on average, the frequencies sum up to more than 100%.

Code	Meaning	Count (Frequency)
EF	The exam was fair	102 (30.5%)
HELPFUL	The pre-exam descriptions of problems was helpful	74 (22.2%)
GS	The pre-exam info guided my study	62 (18.6%)
VARTOOBIG	The difficulty variance on exam was too large	62 (18.6%)
MORE	More practice problems are needed	44 (13.2 %)
ES	The exam was secure	37 (11.1 %)
VB	Variance Bad: Having questions of different difficulty is bad	30 (9.0 %)
UNFAIR	The exam was unfair	27 (8.1 %)
PIH	Perfection is Hard: It's difficult to make variants with equal difficulty	25 (7.5%)
DQPC	Having Different Questions Prevents Cheating on exams	24 (7.2%)
COMM	Student discussed exam with friends or knows people who did	23 (6.9%)
PEMF	Pre-exam info makes exam fair	23 (6.9%)
HARD	Exam was hard or student wanted easier exam	19 (5.7%)
CURVE	Desired a curve or balancing difficulty (e.g. equal # of easy and hard questions)	16 (4.8%)
DOBETTER	Student wanted a fair exam	16 (4.8%)

The 15 most frequently applied codes, which represent approximately 20% of all of the codes, are shown in Table 5. From the codes, it can be seen immediately that there are contradictory sentiments expressed (e.g., "exam was fair" and "exam was unfair").

When the coding was complete, the two researchers then independently identified themes from the code book. The two researchers then reconciled themes to produce a final list. The themes are presented below with representative quotes from the student responses. The themes are ordered according to the frequency of their constituent tags. For instance, the "Exams are sufficiently fair" theme is composed of the EF code, among other codes. The EF code was the most common code, so the theme it supported was listed first.

As with the codes, the themes were somewhat contradictory in nature; some of them emphasize the importance of question versions for security and others suggest that versioning should be eliminated. From reading the student responses, these contradictory themes are never expressed by the same student. These are issues that are raised by different sub-populations of students with different preferences when it comes to ensuring security at the cost of variance in exam difficulty. In Section 5, we explore what factors might lead students to express one of these opinions over another.

Theme 1: Exams were sufficiently fair. Many students indicated that they did not believe their performance was substantially impacted by having question variants.

> Small amount of difficulty variance is reasonable

Many students also indicated that perfection was hard when trying to balance security considerations against the need for questions of equal difficulty.

> I realize it's very difficult to keep all variations of the exam the same difficulty and the variation ensures security.

Some students stated that exams were fair, and that their classmates were characterizing the exams as unfair because of their own personal deficiencies, such as not studying enough or not using appropriate study strategies.

> I think your method for this exam is good enough. People complain about this exam because they did not study hard enough.

> As long as the instructor feels they have prepared their students well enough to answer any of the exam questions, regardless of the difficulty, the student should be able to answer. The problem may be time given at that point. If a problem is more difficult the student should be given more time.

Theme 2: Pre-exam information is helpful. Many students appreciated the provided pre-exam information and found it to be beneficial. Students stated that it helped guide their study and also reduced anxiety.

> The [pre-exam information] did take a lot of pressure off of me, and I think that moving forward, this is a great resource.

> The [pre-exam information] gave me an opportunity to narrow my studying and, as a result, I probably did better than if I had not been given these descriptions

Some students also indicated that providing pre-exam information reduced the advantages that cheaters had and made tests more equitable for all students.

> Releasing the pre-exam descriptions increases exam and security and makes the test more fair. Without the pre-exam descriptions, people who take the exam later will know what kinds of questions might be on the exam by asking others. Releasing the types of questions before hand makes it feel more fair.

Many students stated that the pre-exam information mitigated the issue of unequal difficulty of questions.

> I think that by telling us what would be on the programming exam, the variation in difficulty was small enough.

Theme 3: Versioned questions perceived as unfair. Some students indicated that randomly assigned questions were inherently unfair, even if instructors took steps to ensure that the questions were of similar difficulty. These students may not have been adversely affected by the assignment of questions at random, but rather disagreed with the practice in principle.

> [Instructors] should keep same tests so some people that have the harder version won't do worse than people with the easier version"

Some students indicated that in their specific cases, they were disadvantaged by receiving a specific variant: these students believed that they would have performed better if they had been luckier and received a different question.

> I did find it unfair in this last programming exam that people got mirror trees (which is harder to write an algorithm for) compared to writing a stack or queue with a vector.

> Some questions are definitely disproportionately easier than others.

Theme 4: Exams are secure; cheating is prevented. Many respondents appeared satisfied with the exam security precautions. Many who expressed this sentiment believed that cheating was prevented by the presence of multiple question variants.

> I love how secure the exams are, feels like a level playing field, and even if it isn't at least you guys help me feel that it is with your efforts to randomize questions...

Some students indicated that they valued the security from question variation even if it led to some variance in question difficulty.

> I think the variations are good to prevent cheating and shouldn't be removed. The slight variation in difficulty is worth it.

A few students suggested that the complexity of questions was key to exam security.

> I think the exams are already fairly secure...tough to relay a solution because the [programming] questions are relatively complex

> The failure [on] my programming [exam] B is mostly because I emphasized on remembering the exact code implementation in the style of lab... for specific algorithm and cannot respond to the change [on the exam] flexibly

Theme 5: Students are concerned about cheating. Some students indicated that their classmates communicate with friends about exams and share questions with those who haven't taken an assessment yet (i.e., collaborative cheating).

> People do ask others what questions they had

> From what I hear/overhear in conversations, students who take the exam early tend to be very willing to disclose what they know to students taking the exam later.

Students want instructors to take action to prevent collaborative cheating, and there appears to be broad agreement amongst students that having question pools diminishes the benefits of collaborative cheating

> Make sure the exams are equal in difficulty while still being different to prevent cheating

> If there are variations it is okay as I think that preventing cheating is important

Theme 6: Fairness requires further instructor effort. Many students believed that there were additional steps that instructors could take to ensure a fairer testing experience. These students gave suggestions about how to reach parity during testing.

> If practice questions were made available, I feel like there'd be less complaint[s] about difficulty

> I feel that the exams are okay as is, but one possible solution might be to have multiple questions that a student can select from..

Many students suggested corrective actions which could be undertaken by professors after the administration of tests which may have had questions of different difficulty. This group is distinct from the students who recommended actions that could be taken during testing itself.

> Would it be possible to curve certain questions to match other questions' grade distributions?

> Make some set very difficult, and give [a] big curve on that set.

Theme 7: More instructor transparency is needed. A few students indicated that more information was needed from instructors to assuage their concerns about exam fairness.

> Release the score deviation averages, median and mode for different versions so that it would be more clear for the students whether the questions were equally hard or easy

> ...Invite students to review different versions of exams and discuss their difficulty at the end of this semester. This action should at least help future students ...

4.1 Disjoint Subpopulations

A number of the identified codes and themes contradicted each other, with some indicating that the exam was fair and others decrying the unfairness of the exam. A quick review of the coded student responses verified what we recalled from the coding process, that these contradictory codes and themes were largely the result of different sub-populations of the class.

Our observations are consistent with an overall student population that values both exam security and fairness, but when the two are in contention, some students are willing to accept exam security at a loss of some fairness and other students would prefer fairness, even if it requires some loss of exam security. In the next section, we attempt to identify characteristics of the two populations that explain their preferences relating to security and fairness.

5 QUANTITATIVE ANALYSIS

Having posited that there are at least two sub-populations among the surveyed students, one that valued security over fairness and the other that valued fairness over security, we sought to identify these sub-populations and attempt to explain their security/fairness preferences.

We brainstormed two potential differences in the sub-populations that could potentially explain their preferences. First, a recent negative experience (e.g., getting a harder question variant) might lead students to complain about unfairness. Second, students struggling in the class may be more prone to complain about unfairness, while students that are doing well are satisfied with the status quo. We developed the following two testable hypotheses:

H1 Students that express concerns about unfairness are more likely to have received harder question versions on the most recent exam.

H2 Students that express concerns about unfairness are more likely to be lower performing students in the class as a whole.

We used the codes we assigned to student comments to divide the students into sub-populations. Since we had so many codes, many of which were encountered a small number of times, any attempts to do analysis on students with a single code lacked statistical power. As such, we grouped our codes into three categories:

(1) Status-quo: this category includes the codes that suggest that a student is satisfied with the existing exam structure. The codes that make up Theme 1: *Exams sufficiently fair*, Theme 4: *Exams are secure*, and Theme 5: *Students are concerned about cheating* contribute to this category. We characterize the status quo as being biased towards security with consequences, at least on Exam 4, on fairness.

(2) Unfair: this category includes the codes that suggest that a student desires changes in the status quo because of issues related to unfairness. Among others, codes relating to Theme 3: *Versioned questions perceived as unfair*, Theme 6: *Fairness requires further instructor effort*, and Theme 7: *More instructor transparency is needed* contribute to this category.

(3) Neutral: the rest of the codes were orthogonal to the security/fairness contention and were ignored for the rest of this analysis.

Students were assigned to one of four categories as follows:

(1) Status-quo: students whose comments were tagged by one or more Status-quo codes and zero Unfair codes. (130 students)

(2) Unfair: students whose comments were tagged by one or more Unfair codes and zero Status-quo codes. (98 students)

(3) Neutral: students whose comments were tagged with neither Status-quo or Unfair codes. (72 students)

(4) Mixed: students whose comments were tagged by at least one Status-quo and one Unfair code. (33 students)

5.1 H1: Group correlated to question variant?

Tables 6 and 7 show how many students from each category (Status-quo, Unfair, Neutral, Mixed) received each version of Q1 and Q2, respectively. We used the chi-square test for independence to determine if there was an interaction between question variant and sentiment expressed. For question 1 variants, we did not find evidence of a difference in sentiment; the chi-square test statistic was 9.107, *df* = 15 and *p-value*: 0.8719.

Table 6: Distribution of student opinion by variant received for question 1. *Mean is the mean score on that question; the other numbers are counts of student respondents who received that question version.*

Question	Mean	Status-quo	Unfair	Neither	Mixed
queue	86.4	19	18	9	3
stack	88.5	21	16	13	5
clone	75.3	19	18	13	7
find	78.3	22	15	7	4
insert	74.0	21	14	16	9
mirror	64.3	28	16	14	5

Table 7: Distribution of student opinion by variant received for question 2. *Mean is the mean score on that question; the other numbers are counts of student respondents who received that question version.*

Question	Mean	Status-quo	Unfair	Neither	Mixed
even_pre	44.5	24	24	9	2
neg_pre	44.3	18	13	17	4
nonnull_io	37.1	15	13	19	6
odd_io	35.1	18	17	11	11
pos_io	50.3	23	15	11	4
pos_pre	50.1	32	15	5	6

For Q2 questions, however, we did have a statistically significant relationship between variant received and to which category a student was assigned. The chi-square test statistic was 32.314, *df* = 15 and *p-value*: 0.0058. We performed a post-hoc analysis using the two-sample t-test to compare the sentiment distribution for each pair of Q2 variants. We used the Bonferroni correction to avoid an accidental finding of significance. We found that `pos_pre` had a statistically significantly different distribution of sentiments from `nonnull_io` with a p-value of 0.002.

In spite of the difference observed for Q2, we conclude that students' expressions of unfairness are not motivated by recently receiving one of the harder question versions. We failed to find any relationship with the particular Q1 variant received, and Q1 is the pool where we found a statistically significant difference in difficulty. For Q2, the statistically significant variation in sentiment seems to be between variants unrelated to question difficulty. Some of the easiest question variants in each pool were among the ones that had the highest number of students in the Unfair group.

5.2 H2: Group correlated with performance?

We used the final point tallies in the class at the end of the semester to explore if there was a performance difference between the student classifications. A maximum of 1,000 points could be earned through normal class activities; we did not include extra credit activities for our analysis. The average scores for each group are plotted along with their 95% confidence intervals in Figure 5.

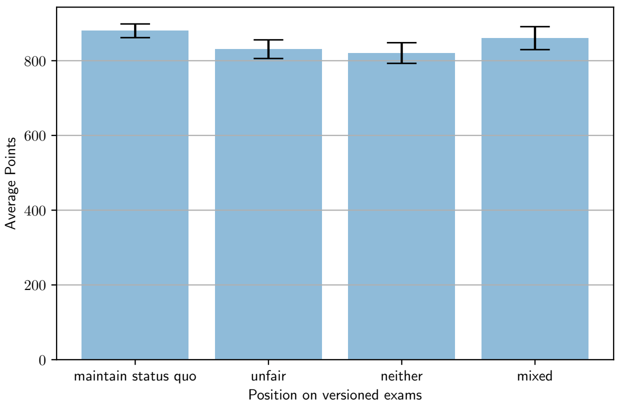

Figure 5: Average Points Based on Sentiment

Table 8: Post Hoc Tests for Scores Based on Sentiment

Groups Compared	Test statistic (t)	P-value
Status-quo vs. Unfair	9.85	0.0019
Status-quo vs. Mixed	0.804	0.3711
Status-quo vs. Neither	12.37	0.00053
Mixed vs. Neither	2.73	0.102
Neither vs. Unfair	0.1883	0.6649
Mixed vs. Unfair	1.5822	0.2107

We used the one-way ANOVA test to identify if there was a statistically significant relationship between the students' sentiments related to the security vs. fairness tradeoff and their overall grades. The F statistic was 5.245, with df = (3, 322) and *p-value* = 0.0015, indicating significance. Given significant omnibus test results, we conducted post-hoc two-sample t-tests for each pair of sentiment groups. This allowed us to detect which specific groups were different. As Table 8 and Figure 5 indicate, Status-quo category students out-perform students in the Unfair and Neither categories by a statistically significant degree.

6 DISCUSSION

Among the survey respondents, we found some students that seem to be more security minded, satisfied with the implementation of the exam and its use of question variants. We also found other students that had significant concerns related to their perception of the exam being unfair. When first pondering why some students would express concerns about fairness and others would not, our first hypothesis was that those that were complaining were the one harmed by the difficulty variance because they had received one of the more harder variants.

We found no support for that hypothesis in our data. Instead, we found a statistically significant correlation between students expressing concerns about unfairness and overall course performance, with students who expressed unfairness concerns performing lower on average than students that were satisfied with the (security-biased) status quo.

We suggest two hypotheses for why lower performance is correlated to concerns of unfairness, one more charitable than the other. The first explanation relates to Covington's Self-Worth Theory [16], which theorizes that people are primarily driven by the need to perceive themselves as competent. This theory suggests that individuals engage in a number of self-protective strategies to maintain a positive self-concept, including "excuse-giving" like attributing failure to uncontrollable factors. Question variants represent an external, uncontrollable, and unstable factor in exams, making them a convenient scapegoat for struggling students. This hypothesis also relates to the hedonistic bias literature [38], which suggests that people have a tendency to take credit for successes, but attribute negative outcomes to external factors.

The second explanation is that lower performing students are more likely to fixate on unfairness because they are more susceptible to it. A high-performing student is likely to achieve a high score on an exam no matter which question variants they receive; difficulty variance between question versions has little influence on their scores. In contrast, lower performing students may be able to complete an easier problem, but fail to complete a harder problem. In fact, if the difficulty variance occurs right around their ability level, even a small difficulty variance could be magnified into a large impact on their grade.

Given that expressing concerns about unfairness is correlated to lower performance in the class, it is perhaps not surprising that expressing concerns about unfairness is not correlated to receiving that hardest problem. A student who is struggling in the class may be unable to correctly gauge the relative difficulty of the problem variants, especially for problems that the instructor anticipated as being of similar difficulty.

One component of the exam that many students were pleased with was the pre-exam information given, believing that it helped reduce instances of collaborative cheating while mitigating potential harm of having multiple question variants on a test. These students characterized the pre-exam information as guiding study (GS), reducing anxiety (RA), or making the exam more fair (PEMF). Based on the responses recorded, we believe that this may be a promising tool that warrants further study. Indeed, previous research showed that students prefer teaching practices that assist them in preparing for exams to practices like curving or other score manipulations that artificially raise grades after a test has been administered. [22].

7 LIMITATIONS

A major limitation of this study was that the data was originally collected with a goal of determining student satisfication with regards to the general administration of tests in the data structures course. It was only after preliminary analysis that we discovered some students characterized the exam as unfair. Hence, the survey questions students received may not have been optimal at eliciting all students' opinions about exam security and exam fairness tradeoffs.

Second, we noted that the survey respondents performed substantially better in the course than non-respondents. It is possible that this group of non-respondents may have expressed opinions that were systematically different from what we received from the rest of the class, which could have affected our results.

Students' comments related to Exam 4 as a whole, so it was generally difficult to determine which of the two programming questions any unfairness concerns were directed. As such, we could not completely isolate sentiments expressed for a variant of the first question from sentiments expressed for a variant of the second question.

8 CONCLUSION AND FUTURE WORK

This paper provides an in depth exploration of students perceptions of the relationship between exam security and fairness in the context of question variants. We find that different sub-populations of students express contradictory opinions of the appropriate set point in the trade-off between exam security and fairness. Some students value the exam security derived from versioned exams to the point that they are willing to tolerate difficulty variance between versions and suggest that students complaining about unfairness did not study enough. For other students, fairness is paramount and they find even the suggestion of question versions to be unfair.

We suspect that there is in fact a continuum of opinions where the above opinions represent the extremes on this continuum. For future work, we would like to understand this continuum better and specifically understand how to address the concerns of the students not satisfied with the status quo. We anticipate that higher fidelity protocols like interviews and/or focus groups are more appropriate for such an emotionally charged issue.

Finally, we believe that question versioning will only become more prominent in computer science instruction. The advantages of computer-based exams for introductory programming courses are compelling, but large enrollments make having all the students solve the same version at the same time challenging to perform securely. In addition, question versioning is a natural tool for online assessments, especially ones which are unproctored.

ACKNOWLEDGMENTS

This work was partially supported by NSF DUE-1915257 and the Grainger College of Engineering at the University of Illinois at Urbana-Champaign under the Strategic Instructional Initiatives Program (SIIP). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] J Stacy Adams. 1965. Inequity in social exchange. In *Advances in experimental social psychology*. Vol. 2. Elsevier, 267–299.

[2] DeLores Rice Brown. 1984. *A comparison of student attitudes and perceptions regarding academic dishonesty of selected class groups in 1980 and 1983 at Iowa State University*. Ph.D. Dissertation. Iowa State University.

[3] Mary Elaine Califf and Mary Goodwin. 2002. Testing Skills and Knowledge: Introducing a Laboratory Exam in CS1. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education (SIGCSE '02)*. ACM, New York, NY, USA, 217–221. https://doi.org/10.1145/563340.563425

[4] Tracy Camp, W. Richards Adrion, Betsy Bizot, Susan Davidson, Mary Hall, Susanne Hambrusch, Ellen Walker, and Stuart Zweben. 2017. Generation CS: The Mixed News on Diversity and the Enrollment Surge. *ACM Inroads* 8, 3 (July 2017), 36–42. https://doi.org/10.1145/3103175

[5] Jacabo Carrasquel, Dennis R. Goldenson, and Philip L. Miller. 1985. Competency testing in introductory computer science: the mastery examination at Carnegie-Mellon University. In *SIGCSE '85*.

[6] A. T. Chamillard and Kim A. Braun. 2000. Evaluating Programming Ability in an Introductory Computer Science Course. In *Proceedings of the Thirty-first SIGCSE Technical Symposium on Computer Science Education (SIGCSE '00)*. ACM, New York, NY, USA, 212–216. https://doi.org/10.1145/330908.331857

[7] A. T. Chamillard and Jay K. Joiner. 2001. Using Lab Practica to Evaluate Programming Ability. In *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education (SIGCSE '01)*. ACM, New York, NY, USA, 159–163. https://doi.org/10.1145/364447.364572

[8] Kathy Charmaz and Linda Liska Belgrave. 2007. Grounded theory. *The Blackwell encyclopedia of sociology* (2007).

[9] Binglin Chen, Matthew West, and Craig Zilles. 2018. How much randomization is needed to deter collaborative cheating on asynchronous exams?. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. 1–10.

[10] Leon Chen. 2019. CS 61A course enrollment reaches an all-time high at 2,000 students. *The Daily Californian* (Sep 2019). https://www.dailycal.org/2019/09/10/cs-61a-course-enrollment-reaches-an-all-time-high-at-2000-students/

[11] Zhuojun Joyce Chen. 2000. The impact of teacher-student relationships on college students' learning: Exploring organizational cultures in the classroom. *Communication Quarterly* 48, 2 (2000), Q76.

[12] Rebecca M Chory and James C McCroskey. 1999. The relationship between teacher management communication style and affective learning. *Communication Quarterly* 47, 1 (1999), 1–11.

[13] Rebecca M Chory-Assad. 2002. Classroom justice: Perceptions of fairness as a predictor of student motivation, learning, and aggression. *Communication Quarterly* 50, 1 (2002), 58–77.

[14] G.J. Cizek. 1999. *Cheating on Tests: How To Do It, Detect It, and Prevent It*. Taylor & Francis.

[15] Computing Research Association. 2017. Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006. https://cra.org/data/Generation-CS.

[16] Martin V Covington. 1984. The motive for self-worth. *Research on motivation in education* 1 (1984), 77–113.

[17] Paul Denny, Sathiamoorthy Manoharan, Ulrich Speidel, Giovanni Russello, and Angela Chang. 2019. On the Fairness of Multiple-Variant Multiple-Choice Examinations. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 462–468.

[18] John Dermo. 2009. e-Assessment and the student learning experience: A survey of student perceptions of e-assessment. *British Journal of Educational Technology* 40, 2 (2009), 203–214.

[19] Lena Feinman. 2018. *Alternative to Proctoring in Introductory Statistics Community College Courses*. Ph.D. Dissertation. Walden University.

[20] Daniel T. Fokum, Daniel N. Coore, Eyton Ferguson, Gunjan Mansingh, and Carl Beckford. 2019. Student Performance in Computing Courses in the Face of Growing Enrollments. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 43–48. https://doi.org/10.1145/3287324.3287354

[21] Emily A. Ford. 2015. *Faculty and Student Attitudes and Perceptions of Academic Dishonesty*. Ph.D. Dissertation. Baker University.

[22] Michael E Gordon and Charles H Fay. 2010. The effects of grading and teaching practices on students' perceptions of grading fairness. *College Teaching* 58, 3 (2010), 93–98.

[23] Valerie J Haines, George M Diekhoff, Emily E LaBeff, and Robert E Clark. 1986. College cheating: Immaturity, lack of commitment, and the neutralizing attitude. *Research in Higher education* 25, 4 (1986), 342–354.

[24] Norman Jacobson. 2000. Using On-computer Exams to Ensure Beginning Students' Programming Competency. *SIGCSE Bull.* 32, 4 (Dec. 2000), 53–56. https://doi.org/10.1145/369295.369324

[25] James Q Knowlton and Leo A Hamerlynck. 1967. Perception of deviant behavior: A study of cheating. *Journal of Educational Psychology* 58, 6p1 (1967), 379.

[26] Herbert W Marsh and Jesse U Overall. 1980. Validity of students' evaluations of teaching effectiveness: Cognitive and affective criteria. *Journal of educational Psychology* 72, 4 (1980), 468.

[27] Donald L McCabe, Linda Klebe Treviño, and Kenneth D Butterfield. 2001. Cheating in academic institutions: A decade of research. *Ethics & Behavior* 11, 3 (2001), 219–232.

[28] An Nguyen and Colleen M. Lewis. 2020. Competitive Enrollment Policies in Computing Departments Negatively Predict First-Year Students' Sense of Belonging, Self-Efficacy, and Perception of Department. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 685–691. https://doi.org/10.1145/3328778.3366805

[29] Virginia P Richmond and James C McCroskey. 1984. Power in the classroom II: Power and learning. *Communication Education* 33, 2 (1984), 125–136.

[30] Rita Cobb Rodabaugh. 1996. Institutional commitment to fairness in college teaching. *New Directions for teaching and learning* 1996, 66 (1996), 37–45.

[31] Lawrence Rudner. 2010. Implementing the Graduate Management Admission Test Computerized Adaptive Test. In *Elements of adaptive testing*, Wim J van der Linden and Cees AW Glas (Eds.). Springer, 151–165.

[32] Mika Saari and Timo Mäkinen. 2016. Utilizing Electronic Exams in Programming Courses: A Case Study. In *EDULEARN16 Proceedings : 8th International Conference on Education and New Learning Technologies (EDULEARN proceedings)*. 7155–7160. https://doi.org/10.21125/edulearn.2016.0560

[33] Mehran Sahami and Chris Piech. 2016. As CS Enrollments Grow, Are We Attracting Weaker Students?. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 54–59. https://doi.org/10.1145/2839509.2844621

[34] Mark J Stehlik and Philip L. Miller. 1985. *Implementing a Mastery Examination in Computer Science*. Technical Report CMU-CS-85-175. Carnegie Mellon University.

[35] Anselm Strauss and Juliet Corbin. 1998. *Basics of qualitative research techniques*. Sage publications Thousand Oaks, CA.

[36] Elaine Walster, G William Walster, and Ellen Berscheid. 1978. Equity: Theory and research. (1978).

[37] Michael P. Watters, Paul J. Robertson, and Renae K. Clark. 2010. Student perceptions of cheating in online business courses. *Journal of Instructional Pedagogies* 6 (2010).

[38] Gifford Weary. 1979. Self-serving attributional biases: Perceptual or response distortions? (1979).

[39] Bernard E Whitley. 1998. Factors associated with cheating among college students: A review. *Research in Higher Education* 39, 3 (1998), 235–274.

[40] Jennifer Yardley, Melanie Domenech Rodríguez, Scott C Bates, and Johnathan Nelson. 2009. True confessions?: Alumni's retrospective reports on undergraduate cheating behaviors. *Ethics & Behavior* 19, 1 (2009), 1–14.

[41] C. Zilles, R. T. Deloatch, J. Bailey, B. B. Khattar, W. Fagen, C. Heeren, D Mussulman, and M. West. 2015. Computerized Testing: A Vision and Initial Experiences. In *American Society for Engineering Education (ASEE) Annual Conference*.

[42] Craig Zilles, Matthew West, Geoffrey Herman, and Timothy Bretl. 2019. Every university should have a computer-based testing facility. In *Proceedings of the 11th International Conference on Computer Supported Education (CSEDU)*.

[43] Craig Zilles, Matthew West, David Mussulman, and Timothy Bretl. 2018. Making testing less trying: Lessons learned from operating a Computer-Based Testing Facility. In *2018 IEEE Frontiers in Education (FIE) Conference*. San Jose, California.