A Multi-Institutional Assessment of Oral Exams in Software Courses

Peter Ohmann
College of Saint Benedict / Saint John's University
Saint Joseph, MN, United States
pohmann001@csbsju.edu

Ed Novak
Franklin and Marshall College
Lancaster, PA, United States
enovak@fandm.edu

ABSTRACT

Oral exams are an inviting alternative to traditional paper-and-pencil exams. However, they are largely under-utilized in computer science education. In this report, we describe our design for comprehensive final oral exams in five software engineering class sections, across two different small institutions. We present our exam format and our subjective assessment of the exam format in assessing student knowledge as instructors. We also gather quantitative and qualitative data from student surveys. We surveyed students before and after the oral exam to assess their perceptions of it, including their predicted grade and their subjective opinions and experiences. Our work shows evidence that oral exams are effective and practical mechanisms for software engineering classes of a smaller size (approximately 20 students). Student survey responses indicated favorable feedback for our oral exam format; students viewed oral exams as a good assessment of their knowledge and useful beyond that individual class.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; *Software engineering education*; **Student assessment**.

KEYWORDS

oral exam; software engineering education; student survey

ACM Reference Format:
Peter Ohmann and Ed Novak. 2025. A Multi-Institutional Assessment of Oral Exams in Software Courses. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701848

1 INTRODUCTION

Assessing student knowledge is a critical part of education, and written exams are a common means of assessing comprehensive knowledge from a course (e.g., as a "final exam"). Oral exams are an alternative where students meet individually with their instructor; these exams promise flexibility in how students can demonstrate their knowledge of topic areas. While interest and research on oral exams is increasing, they are still severely understudied in computer science education. Studies analyzing student views on oral exams across various types of courses are especially lacking. It is imperative, then, to explore the strengths and weaknesses of oral exams in modern CS education, and the emotional experiences of students taking oral exams versus other assessment methods.

Where many find traditional exams strict and formal, oral exams present a uniquely social and human alternative. Oral exams also deter academic dishonesty, even in light of recent LLM advancements, since they are carried out with the full attention of the instructor. Finally, oral exams are relevant outside of academia due to their similarity to many workforce tasks (e.g., discussing progress with a superior or interacting with a client) and technical interviews.

However, oral exams are difficult to design and administer well. The improvisational nature of oral exams can lead conversations off-topic and also lead to inconsistency between exam-takers. Because oral exams are not common, most educators are not experienced with them and don't feel confident writing or designing an oral exam that will be fair, effective, enjoyable, and relevant to students.

In this paper, we present our experiences as instructors at two US-based colleges, using oral exams in place of traditional written exams for final exams in software development courses. We also gather student survey data (approximately 60 responses from 5 sections across the 2 institutions) to measure student perceptions of the oral exam format. Our survey reveals some interesting aspects of oral exams including generally positive student sentiment, desire for more oral exam opportunities, and grade confidence differences by gender. The key components of this report are:

- Design and implementation reflections for oral exams conducted across multiple sections of software courses at two different institutions by different instructors.
- Analysis of pre- and post-surveys of student perceptions regarding these oral exams.
- Instructor insights from our experiences conducting these oral exams, and reflection on study survey data.

2 RELATED WORK

Limited prior work exists studying the use of oral examinations in computer science education. In fact, before the doctoral level [24], oral exams are rare in most post-secondary disciplines in the United States [6, 10] outside of foreign language courses [3, 8]. Despite limited work to this point, interest in oral assessment is growing.

Our prior work [16] presented the exam format we adapt in this paper, chosen for its flexibility to adapt to our software-development course setting, and the ability to administer substantively the same exam to all students (see section 3). In that work, we also compare the effectiveness of final oral exams and final written exams in an

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0531-1/25/02…$15.00
https://doi.org/10.1145/3641554.3701848

introductory computing course. Findings align with other prior work, including reduced time pressure, flexibility in correcting early problem errors, and benefits in assessing student learning and problem-solving strategies [5, 21] but also equity concerns and issues scaling exam time for larger class sizes [6, 13, 21]. We thus expand the use of our prior exam format into a new class setting, and provide evaluation of a more extensive implementation: across two institutions over three semesters in five total class sections.

Other work on oral exams is less directly related, but still informs our design and implementation. Oral exams, particularly recorded video, gained some popularity (and, hence, prompted some study) during remote learning [11, 12, 15, 22, 25]. While our work here may also apply in the remote setting, our evaluation focuses on designing and evaluating student perceptions of the in-person setting. Others, including Reckinger and Reckinger [19, 20], specifically address issues scaling oral laboratory assessment to large course sizes, including use of TA support. We examine some similar issues, including self-efficacy on oral exams for underrepresented groups, but target a different type of exam (summative final exams). Related work by Asklund and Bendix [1, 2] also describes oral evaluation in a software-related course, though that course focuses more on software configuration management, and the oral evaluations are designed to assess group performance. Similarly, Porquet-Lupine and Brigham [18] present an approach for evaluating group work using oral assessment coupled with peer evaluations. Other works by Asklund and Bendix [2] and Gharibyan [5] offer recommendations for conducting oral exams; this work is much in line with (and partially inspired) the exam format we adopt [16]. Finally, oral exams have seen some recent study in other STEM disciplines, including mathematics [9], physics [26], and chemistry [4, 23].

3 APPROACH

This section describes our oral exam design, and differences for our two institutions. We closely followed our previous work's exam mode [16], and so only briefly describe the details here. We instead emphasize the unique challenges and opportunities in oral exam design for our two software development courses, which had similar learning outcomes but differing formats.

3.1 Course Descriptions

Both institutions are private, liberal arts, primarily-undergraduate institutions. Both courses are software development courses, cover core topics such as software design and version control, and require a semester-long project in teams of 4–6 students.

Institution 1 (CSB/SJU)'s offering is an upper-division course taken mostly by CS majors and minors, and introduces students to software development. In the course, students develop a desktop application and then move into "full-stack" web development.

The course taught at Institution 2 (F&M) is designed primarily as an elective for CS majors and minors in their third or fourth year of study. The course covers software engineering from the perspective of creating Android applications for external clients.

3.2 Exam Design

The exam is structured as a one-on-one in-person conversation. In essence, the instructor reads each exam question, listens to the student's response, and uses "checkboxes" to award points. As noted in our prior work [16], oral exams need to be *flexible*, but also *consistent* and *equitable*. This means that each student receives substantively the *same exam*, but can ask clarifying questions and/or demonstrate knowledge of a topic in a variety of ways (e.g., while speaking is usually the default mode, written answers, images, and whiteboard work are acceptable and often useful). In practice, this means that the exam rubric is a *strict script* that states everything the exam proctor is *allowed to say*, outside of answering clarifying questions or providing definitions as requested by students.

The difficulty of defining "clarifying questions" is significant. It is important that the instructor is careful not to give any students an advantage on the exam, so comments that hint at correct answers should be avoided. At the same time, one advantage of oral exams is to make the exam accessible and comfortable for every student. Conversing with each student is a balancing act.

Finally, while our format is designed with face-to-face exams in mind, it works equally well for virtual exams (e.g., using video-chat technology such as Zoom or Google Meet). In fact, one student from F&M in our study required a virtual exam due to health concerns.

3.3 Question Design

Recall that our questions must serve as both a script for the test administrator and a rubric to grade student responses. Students are generally *not* expected to check every box to earn a perfect score (since the exam provides *multiple* ways to demonstrate knowledge). In both courses, the oral exams were structured around three high-level topics:

- GitHub / git version control
- Platform-specific software engineering (Android development or client+server web development in our experiments)
- Questions based on existing code such as unit testing and team project contributions

Figure 1 shows simplified exemplary questions used in our exams. Of course, on the actual exam there were more questions and lengthier questions. These questions also highlight important design decisions we made when collaboratively designing oral exams for two similar courses, but with important differences. Each question is designed with a "tell me about X" guiding principle, giving the student an opportunity to express their knowledge about each concept or topic in a more open-ended way.

Figure 1a shows a question used by both instructors on version control. Note the parentheticals, which indicate optional comments that the instructor may read. These are important to orient some students, save time when the exam is running slowly for other students, or to prompt when a follow-on question relies on certain assumptions from that previous question.

Recall that the development platform differed at each institution. We found our design easy to adapt to different contexts. Figure 1b shows an example of how we adapted (the start of) questions to our unique needs. We had similar goals in figure 1c, where students had to analyze provided code using skills from the course. The first author (CSB/SJU) specifically wanted to have students demonstrate test-design knowledge using code that was *similar but distinct* from their project code, while the second author (F&M) wanted to have students answer questions about *their group's* project code.

Version Control Question
We used git + github version control for managing our code this semester. Why was version control important?
☐ Answers question
(How about as an individual?)
☐ Accessing code history ☐ Reverting changes
(How about as a team?)
☐ Sharing code ☐ Resolving conflicts

(a) GitHub / git version control

Web Development Question (CSB/SJU)
The web is "request/reply" or "client/server" oriented. What does it mean that "everything is requests and replies"?
☐ Answers question
☐ Loading pages = requests and/or HTML is reply
☐ Mention no saved state between page loads / cookies

Android App Development Question (F&M)
In Android app programming there is no `main()` what general design or system replaces it?
☐ Answers question
☐ Mentions `onCreate()` method
☐ Other "on" methods: `onPause()`, `onResume()` etc.
☐ Mentions keyword: "Event Driven" architecture

(b) Platform-specific software engineering

Unit Testing Question (CSB/SJU)
[Show code for `UserWallet` Java class]
I want to do black-box testing of the `createUser()` method. Walk me through the process: identify equivalence classes, and determine your test cases.
☐ Answers question (even if not correct)
(Use the board/paper to demonstrate, and talk through it.)
☐ Mostly correct equivalence classes
☐ Make test cases without prompting

Team Contribution Question (F&M)
[Pull up a Java class from their group project. Pick a method / segment that is approximately 20-40 lines long.]
Can you walk me through this section of code? At a high level what does this code do?
☐ Answers question (even if not correct)
☐ Gives high-level description / purpose of code
How could this section of code be improved (if at all)?
☐ Answers question
☐ Improved efficiency, maintainability, bug fix, etc.

(c) Existing code questions

Figure 1: Sample Oral Exam Questions and Rubric; prompts abbreviated for space

3.4 Scheduling, Running, and Grading Exams

The oral exams at both institutions were scheduled one-on-one meetings between the instructor and students. At F&M, direct email and typical face-to-face communication was sufficient to schedule each exam slot with each student, with specific slots in each general window of availability allocated on a first-come-first-served basis. At CSB/SJU, due to the larger number of students, the instructor used a digital scheduling system, where students were assigned "registration times," much like for semester registration.

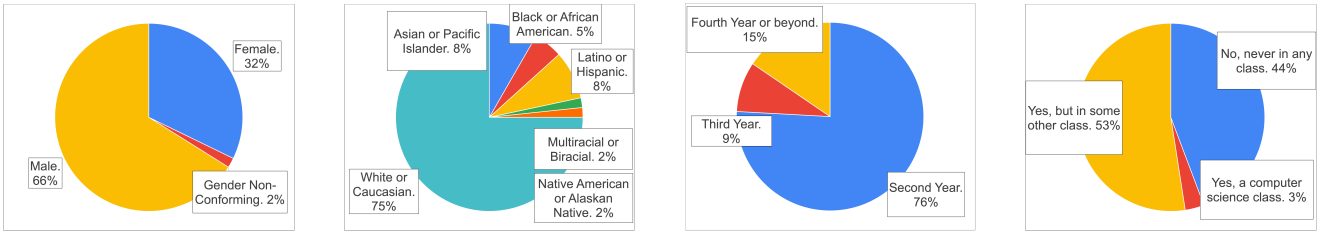
At many institutions a three hour exam block is scheduled for the entire class with the presumption that the course will have a traditional synchronous paper-and-pencil exam. While it was not an issue at the institutions in this study, we recognize that some institutions may object to instructors scheduling different exam times for different students spread across several days.

Each oral exam is designed to take 20 minutes, but is booked in a 30 minute time slot. For students with accessibility and/or disability accommodations (like 50% extra time on tests), the appointment was adjusted accordingly. To help students manage time, instructors gave warnings if students took a disproportionate amount of time on any one question, and near the end of the time slot (5-7 minutes before). In our experience, very few students used their full time slot, even without prompting. Many had time to revisit previous question(s) later in their time slot.

Because of the in-exam check-boxes and one-on-one nature of the oral exam, grading is essentially complete when the student finishes the exam. The instructor can *immediately* give them an indication of their score, and within just a few minutes can calculate their precise score. We utilize this fact to analyze student confidence in our evaluation (see section 4), but it might also give instructors flexibility to estimate grades but later adjust the grade if necessary (e.g., if an invalid question is later discovered). That said, rubric design is very important for this particular oral exam design, and we did not need to make adjustments at either of our institutions.

Institution	Semester	Sections	Pre-Survey Response	Post-Survey Response	Grade Estimate Data
CSB/SJU	Spring 2023	2	30 / 40	25 / 40	32 / 40
F&M	Fall 2023	1	—	8 / 16	—
CSB/SJU	Spring 2024	2	23 / 40	19 / 40	24 / 40

Table 1: Course Survey Response Data. Response numbers indicate the number of complete responses to the survey (used in analysis) versus the number of students in that class semester.



(a) Gender (free text available) **(b) Race/Ethnicity (free text available)** **(c) Academic year** **(d) Previous oral exam**

Figure 2: Self-reported student demographics; percentages reflect responses to each (optional) question in the survey.

4 ASSESSMENT

In this section, we describe our assessment of the oral exam design and applicability for later-year software development courses. First, we analyze survey data gathered across our two institutions. Then, we reflect on this survey data and our own subjective assessment as instructors to draw conclusions and determine practical recommendations based on our results.

4.1 Survey Analysis

Table 1 summarizes the survey data we gathered to assess our implementation. Institution 1 (CSB/SJU) conducted online pre-surveys one week prior to the final exam as well as post-surveys via a link sent to students after their individual exam. Institution 2 (F&M) conducted surveys only after the oral exam, but used the same set of post-survey questions. The first author (CSB/SJU) also asked students to voluntarily estimate their exam grade (see results later in this section). All surveys were anonymous, using an anonymous identifier to link pre and post surveys with estimated and actual exam scores. All sections of both courses taught at CSB/SJU were by the first author, and the section at F&M was by the second author.

Figure 2 shows demographic data for our survey respondents. Demographic questions were optional, so numbers for future analysis based on these characteristics may not reflect the entire population. As noted in section 3, the CSB/SJU course primarily serves second- and third-year students, while the F&M course serves juniors and seniors. Per figure 2d, students were about evenly split in whether they had taken any sort of oral exam, but only one student reported taking an oral exam in computer science previously. We also asked students about their intended major of study. Not surprisingly, given that that these were upper-division courses, all were computer science students (with the vast majority majoring or double-majoring in computer science), so we do not report those results here.

As we later discuss in detail, results in this section are not statistically significant due to our small sample sizes and response rates relative to course sizes. Nevertheless, the results provide interesting preliminary data to support further investigation into the effectiveness of oral exams in the CS community more broadly.

Table 2 shows survey results, as mean values across both institutions and all sections. All questions are Likert scale (1-5), and some are abbreviated here for space[1].

Responses across all positive questions (Q1, Q2, Q5, Q6, Q8, Q9, and Q10) were high (i.e., closer to 5). While a moderate number of students indicated that they were nervous for the oral exam (Q3), the only student comment on this point indicated "a good kind of nervous for the oral final" (CSB/SJU). We were especially encouraged to see that students felt the exam was equitable (Q8), and was a better assessment of their knowledge than a traditional written exam (Q5). Note that the pre-survey version of Q4 asked students to compare their expected performance to a hypothetical written exam, while the post-survey compared to their "expectation prior to the exam," so any value above 3.0 is a *better than expected* result. We were also happy to see an increase in Q6 (usefulness beyond this class), indicating that students saw their exam not only as an assessment, but practice for future life skills. (This increase manifested each semester within CSB/SJU, as well.)

We also asked students, on their post-survey, whether they thought oral exams should be offered or required in other CS courses; figure 3 shows responses. More than half of respondents thought that *all* courses should at least offer oral exams as an option, and no respondents indicated that oral exams should be excised from all courses (which would have indicated a strong negative experience). Our goal here was not to indicate that all courses should use this style of exam, but rather that students are *excited* to see expanded use of this methodology, even if it is not a perfect fit for all courses.

Finally, we analyzed the data from CSB/SJU on student grade estimates. As indicated earlier, we asked students for their estimate

[1]Unless otherwise specified, 1=Strongly Disagree and 5=Strongly Agree. Full versions of all questions and histogram data are available in our technical report [17].

Question (Likert (1-5))	Pre-Survey Mean	Post-Survey Mean
Q1: I feel prepared for my final exam.	3.981	–
Q2: I am excited for taking an oral final exam.	3.755	–
Q3: I am nervous about taking an oral final exam.	3.094	–
Q4: Expected grade compared to written/predicted. (5=much higher)	3.660	3.538
Q5: Assessment of knowledge compared to written. (5=much better)	3.792	3.865
Q6: An oral exam will be useful to me beyond this class.	4.170	4.269
Q7: Difficulty compared to (written) midterm exam. (5=much harder)	–	2.135
Q8: The oral exam was conducted fairly and equitably.	–	4.692
Q9: I was satisfied with my performance on the exam.	–	4.173
Q10: I would be excited to take another oral exam.	–	4.231

Table 2: Survey results; some questions were asked in both pre and post surveys

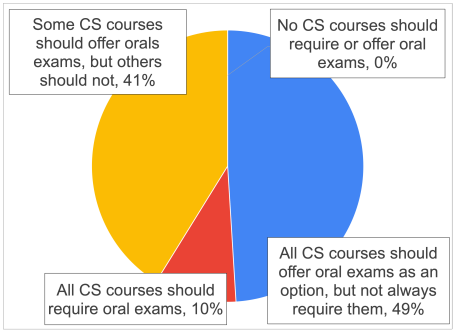


Figure 3: Q11: Should future computer science courses offer oral exams as a requirement or an option?

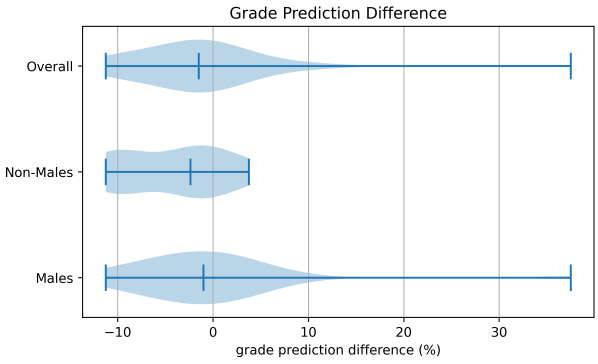


Figure 4: Difference in predicted grade versus actual grade

of their grade immediately after conducting the exam (at the end of the student's exam time). Students were not required to provide an estimate (see table 1), and were allowed to estimate using any format they liked. If a student gave a non-numeric estimate (as was common), we translated the estimate into the closest percentage. For example, if a student answered "an A or B," we treated that response as 90% (precisely on the border); if a student answered "B," we used 85% (exactly the middle of that grade level).

Figure 4 shows results. The values here indicate the difference between the student's estimate and their actual score (out of 100). Thus, a 0 indicates a perfect estimate, while a −10 indicates an underestimate by one full grade level (10%). Overall, students tended to very slightly underestimate their score. However, when we analyzed results by gender identity, we found that non-male students tend to show much lower confidence in their performance (mean −4.4 versus −0.7 for male-identifying students). This result is in line with prior research [7, 14], and suggests the need for continued work on promoting belonging and confidence for all students in CS. (We did not separately analyze all gender identities or report results on racial identity to avoid de-anonymizing individuals.)

4.2 Survey Comments

Students were also allowed to provide any comments to support or clarify their responses. Some clear themes emerged. First, students especially enjoyed the conversational nature of the exam, the ability to ask clarifying questions, and the opportunity to elaborate on initially-unclear answers (13 students / 27 total comments).

> "Explaining a concept in person is much easier than on paper, as you can get immediate feedback and explain and clarify things that you would not normally recognize as needing clarification." (CSB/SJU)

> "I really liked the idea of sitting down and talking to my prof to show my knowledge. I felt like they were able to gauge what I knew better than a written exam." (CSB/SJU)

> "The accessibility of clarification was very helpful." (CSB/SJU)

> "It wasn't just pure memorization—[my instructor] was expecting us to relate to our probelms and expand on our answers." (F&M)

> "I found the exam relatively easy but felt it was more reflective of my knowledge than written exams typically are. Written exams I feel are much more vulnerable to miscommunication, and have a much stronger time pressure relative to the amount of content that can be assessed." (F&M)

That said, not all students felt as comfortable engaging in conversation and asking questions (2/27), or believed that the format may not be general (2/27).

> "[...] it's easier to completely blank on something when the instructor asks a question, even when it's something I know." (CSB/SJU)

> "[...] it's really variable depending on how comfortable a student is meeting with a professor." (F&M)

In keeping with prior findings [16], some students actually indicated *less* stress than a traditional written exam, and saw potential benefits for supporting mental health (3/27).

> "I thought it was a good way to make the exam experience less stressful." (CSB/SJU)

> "I think exams place way too much pressure on someone's mental health especially if something is happening significant in their lives. So there I like idea of one on one with the professors and you get a little time to answer the question without pressure." (F&M)

Perhaps in support of Q6 (table 2), multiple students highlighted relevance or preparation for future technical interviews (4/27).

> "This is great speaking practice for interviews in the future, as being an exam it is able to mimic the high stress of an interview." (CSB/SJU)

> "It is also, in my opinion, a good way to practice common behavioral questions that are asked on job interviews where recruiter picks some project from ur resume and then grills u on it." (F&M)

Finally, most students appeared to appreciate the immediate feedback possible in this exam format.

> "I liked that it was short and that I knew roughly what my grade was as soon as I was done." (CSB/SJU)

> "It was very nice to be done with it [in under] 30 mins compared to the traditional 3 hour exam." (F&M)

Some students, though, indicated some mismatch between their expectations and the actual exam (3/27).

> "I didn't really know what to expect going into it [...] After taking it I feel like I could've done better if I had the knowledge I do now after taking it." (CSB/SJU)

> "I also had no freaking idea how to study for it." (F&M)

Notably, this issue is not unique to the oral exam format, though it would likely be reduced if the format were more prevalent in students' educational experiences (figure 2d). One student whose pre-survey response indicated worry about their ability to think through answers in the format:

> "I am assuming the freedom to slow down and think on a difficult problem is lost in an oral exam." (CSB/SJU)

ended up speaking purely positively of the experience afterwards.

> "Explaining a concept in person is much easier than on paper." (CSB/SJU)

4.3 Recommendations and Threats to Validity

As instructors, we agree with students surveys and found ourselves better able to assess student knowledge in the oral format. Providing students with additional clarification for questions did not feel problematic, and instructors were able to avoid going "off-script," thereby preserving fairness in the exam. However, both instructors found it difficult to determine the "right" time to provide students with additional prompting listed in exam scripts (which, per section 3, generally results in the "loss" of some checkboxes, and, hence, some points).

Cooperatively designing the exam rubric also went smoothly, and both instructors found it especially helpful to get feedback on parallel questions, like those shown in figure 1c. As noted in section 3, the authors took different approaches for having students analyze code (i.e., instructor-authored code similar to project code for CSB/SJU and actual student-written code for F&M). Both approaches seemed to work well, though the second author (F&M) noted that it was, at times, difficult to find student code that clearly met the exam objectives.

In this instance, the timing for proctoring the exam was reasonable, helped significantly by the fact that grading is part of administering the exam. Nevertheless, the first author (CSB/SJU) proctored exams for 40 students each semester, which consumed 20 hours of final exam week each semester. This time commitment is certainly beyond a traditional written exam, even accounting for grading time, and is compressed into a shorter time period (i.e., finals week). Further work à la Reckinger and Reckinger [19], incorporating TA support, would likely be necessary to scale to larger classes. The format we adopt, with strictly-scripted rubrics, should be helpful in this regard.

As noted in section 4.1, our results are not statistically significant due to our smaller sample sizes. However, we gathered results with two instructors across two different institutions and three semesters. Thus, our results (and particularly our recommendations in this section) are applicable for different instructors, and would likely be useful for similar software engineering courses at similar institutions. We acknowledge that our results may not generalize to very different settings, especially much larger course sections.

Evaluating how well students achieve learning objectives with oral exams vs traditional exams was not studied in this work, since it was covered by our prior work [16]. We do see that question as a promising area to be studied thoroughly in future work(s).

5 CONCLUSIONS AND FUTURE WORK

Through our work implementing and gathering data on oral exams in undergraduate software development courses, we found overall positive results. As instructors, we found the workload manageable, and found ourselves better able to assess student knowledge. Students also gave positive feedback via anonymous survey data, indicating that the exams were equitable, felt conversational, and were relevant for future career skills.

Our findings also point out many venues for future work. We found a striking difference in student self-reported grade estimates for non-male-identifying students. This difference is in line with the challenges underrepresented students saw in the work by Reckinger and Reckinger [20], and indicates the need for further research and supports for students in oral exam formats. Overall, though, students indicated a desire for more exams in this vein, and future work should investigate oral exam effectiveness in more areas of computer science.

REFERENCES

[1] Ulf Asklund and Lars Bendix. 2003. Oral vs. written evaluation of students. *LTHs 1: Pedagogisk inspirationskonferens, Lund, Sweden* (2003).

[2] Ulf Asklund and Lars Bendix. 2003. A Software Configuration Management Course. In *Software Configuration Management, ICSE Workshops SCM 2001 and SCM 2003 Toronto, Canada, May 14-15, 2001 and Portland, OR, USA, May 9-10, 2003. Selected Papers (Lecture Notes in Computer Science)*, Bernhard Westfechtel

and André van der Hoek (Eds.), Vol. 2649. Springer, 245–258. https://doi.org/10.1007/3-540-39195-9_18

[3] Barbara Gross Davis. 2009. *Tools for teaching*. John Wiley & Sons.

[4] David E Gardner and Andrea N Giordano. 2023. The challenges and value of undergraduate oral exams in the physical chemistry classroom: A useful tool in the assessment toolbox. *Journal of chemical education* 100, 5 (2023), 1705–1709.

[5] Hasmik Gharibyan. 2005. Assessing students' knowledge: oral exams vs. written tests. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE 2005, Caparica, Portugal, June 27-29, 2005*, José C. Cunha, William M. Fleischman, Viera K. Proulx, and João Lourenço (Eds.). ACM, 143–147. https://doi.org/10.1145/1067445.1067487

[6] M Hairer. approx 2007. Oral examinations/presentations. *Mathematics Research Centre, University of Warwick, UK* (approx 2007). http://homepages.warwick.ac.uk/~masbm/PCAPP/Projects/Oral_exams.pdf

[7] Brian Harrington, Shichong Peng, Xiaomeng Jin, and Minhaz Khan. 2018. Gender, confidence, and mark prediction in CS examinations. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2018, Larnaca, Cyprus, July 02-04, 2018*, Irene Polycarpou, Janet C. Read, Panayiotis Andreou, and Michal Armoni (Eds.). ACM, 230–235. https://doi.org/10.1145/3197091.3197116

[8] Elaine Hewitt and Jean Stephenson. 2012. Foreign language anxiety and oral exam performance: A replication of Phillips's MLJ study. *The Modern Language Journal* 96, 2 (2012), 170–189.

[9] Paola Iannone and Naďa Vondrová. 2024. The novelty effect on assessment interventions: a qualitative replication study of oral performance assessment in undergraduate mathematics. *International Journal of Science and Mathematics Education* 22, 2 (2024), 375–397.

[10] Gordon Joughin. 2010. *A short guide to oral assessment*. Leeds Met Press in association with University of Wollongong.

[11] Rachel S. Lim, Joe Gibbs Politz, and Mia Minnes. 2023. Stream Your Exam to the Course Staff: Asynchronous Assessment via Student-Recorded Code Trace Videos. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education, Volume 1, SIGCSE 2023, Toronto, ON, Canada, March 15-18, 2023*, Maureen Doyle, Ben Stephenson, Brian Dorn, Leen-Kiat Soh, and Lina Battestilli (Eds.). ACM, 144–150. https://doi.org/10.1145/3545945.3569803

[12] Marko Lubarda, Nathan Delson, Curt Schurgers, Maziar Ghazinejad, Saharnaz Baghdadchi, Alex Phan, Mia Minnes, Josephine Relaford-Doyle, Leah Klement, Carolyn Sandoval, and Huihui Qi. 2021. Oral exams for large-enrollment engineering courses to promote academic integrity and student engagement during remote instruction. In *IEEE Frontiers in Education Conference, FIE 2021, Lincoln, NE, USA, October 13-16, 2021*. IEEE, 1–5. https://doi.org/10.1109/FIE49875.2021.9637124

[13] Muhammed Ashraf Memon, Gordon Rowland Joughin, and Breda Memon. 2010. Oral assessment and postgraduate medical examinations: establishing conditions for validity, reliability and fairness. *Advances in health sciences education* 15, 2 (2010), 277–289.

[14] Lena Nekby, Peter Skogman Thoursie, and Lars Vahtrik. 2015. Gender differences in examination behavior. *Economic inquiry* 53, 1 (2015), 352–364.

[15] Ed Novak and Peter Ohmann. 2023. Oral Exams in CS-Education: Pros and Cons in the Age of AI Assisted Programming. *Journal of Computing Sciences in Colleges* 39, 3 (2023), 32–33.

[16] Peter Ohmann. 2019. An Assessment of Oral Exams in Introductory CS. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE 2019, Minneapolis, MN, USA, February 27 - March 02, 2019*, Elizabeth K. Hawthorne, Manuel A. Pérez-Quiñones, Sarah Heckman, and Jian Zhang (Eds.). ACM, 613–619. https://doi.org/10.1145/3287324.3287489

[17] Peter Ohmann and Ed Novak. 2024. *Supplementary Materials to A Multi-Institutional Assessment of Oral Exams in Software Courses*. Technical Report. Franklin and Marshall College. https://digital.fandm.edu/multi-institutionalassessment.

[18] Joël Porquet-Lupine and Madison Brigham. 2023. Evaluating Group Work in (too) Large CS Classes with (too) Few Resources: An Experience Report. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education, Volume 1, SIGCSE 2023, Toronto, ON, Canada, March 15-18, 2023*, Maureen Doyle, Ben Stephenson, Brian Dorn, Leen-Kiat Soh, and Lina Battestilli (Eds.). ACM, 4–10. https://doi.org/10.1145/3545945.3569788

[19] Scott J Reckinger and Shanon Marie Reckinger. 2021. Oral proficiency exams in high-enrollment computer science courses. In *2021 ASEE Virtual Annual Conference Content Access*.

[20] Scott J. Reckinger and Shanon M. Reckinger. 2022. A Study of the Effects of Oral Proficiency Exams in Introductory Programming Courses on Underrepresented Groups. In *SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education, Providence, RI, USA, March 3-5, 2022, Volume 1*, Larry Merkle, Maureen Doyle, Judithe Sheard, Leen-Kiat Soh, and Brian Dorn (Eds.). ACM, 633–639. https://doi.org/10.1145/3478431.3499382

[21] Lee Roecker. 2007. Using oral examination as a technique to assess student understanding and teaching effectiveness. *Journal of Chemical Education* 84, 10 (2007), 1663.

[22] Mihaela Sabin, Karen H. Jin, and Adrienne Smith. 2021. Oral Exams in Shift to Remote Learning. In *SIGCSE '21: The 52nd ACM Technical Symposium on Computer Science Education, Virtual Event, USA, March 13-20, 2021*, Mark Sherriff, Laurence D. Merkle, Pamela A. Cutter, Alvaro E. Monge, and Judithe Sheard (Eds.). ACM, 666–672. https://doi.org/10.1145/3408877.3432511

[23] Carrie R Salmon, Paul A Bonvallet, Yuan Xue, and Susan E Ramlo. 2024. Oral Examinations of Structure and Function in General and Organic Chemistry Courses. *Journal of Chemical Education* 101, 3 (2024), 921–929.

[24] Kathryn E. Sanders and Robert McCartney. 2003. Program assessment tools in computer science: a report from the trenches. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, 2003, Reno, Nevada, USA, February 19-23, 2003*, Scott Grissom, Deborah Knox, Daniel T. Joyce, and Wanda Dann (Eds.). ACM, 31–35. https://doi.org/10.1145/611892.611926

[25] Tammy VanDeGrift. 2022. Post-Exam Videos for Assessment in Computing Courses: See and Hear Students' Thinking. In *SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education, Providence, RI, USA, March 3-5, 2022, Volume 1*, Larry Merkle, Maureen Doyle, Judithe Sheard, Leen-Kiat Soh, and Brian Dorn (Eds.). ACM, 230–236. https://doi.org/10.1145/3478431.3499273

[26] Klára Velmovská and Simona Gorčáková. 2022. Oral examinations of the student during distance physics education. In *AIP Conference Proceedings*, Vol. 2458. AIP Publishing.