ASEE 2022 ANNUAL CONFERENCE

Excellence Through Diversity

MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36460

When is Automated Feedback a Barrier to Timely Feedback?

Andrew Deorio (Lecturer)

Andrew DeOrio is a teaching faculty member at the University of Michigan and a consultant for web and machine learning projects. His research interests are in engineering education and interdisciplinary computing. His teaching has been recognized with the Provost's Teaching Innovation Prize, and he has twice been named Professor of the Year by the students in his department. Andrew is trying to visit every U.S. National Park.

Christina Keefer (University of Michigan)

When is Automated Feedback a Barrier to Timely Feedback?

Christina Keefer, Andrew DeOrio
Department of Electrical Engineering and Computer Science
University of Michigan

1 Abstract

Computing programs have seen a substantial enrollment increases in recent years. One of the challenges brought by rising enrollments is long wait times for students to receive help in office hours. Schools tackling aggressive scaling have turned to peer teaching and automated feedback mechanisms to aid students seeking help.

We examine the relationship between the demand for peer teaching and different automated feedback mechanisms. More specifically, in response to a fixed per capita supply of office hours, do different types of autograder feedback drive different levels of demand for office hours?

Our study examines 3 years of data from 17 computing courses at a large, public university comprising 4 different types of autograder feedback. We collected 105941 unique encounters between students and peer teachers in office hours. Each encounter includes information about the student and staff member involved, the amount of time that the student waited, and the amount of time that the student was helped.

Our results suggest that some automated feedback mechanisms used to provide timely feedback are counter-intuitively associated with greater demand for office hours and longer wait times for students to receive feedback. In particular, more opaque automated feedback was associated with up to 57% longer wait times for office hours compared to more transparent automated feedback. A course that switched from opaque feedback to more detailed feedback decreased wait times by 24%. These results can help instructors leverage automated feedback while ensuring timely access to peer teachers in office hours.

2 Introduction

The past decade has seen an explosion in the number of students enrolled in undergraduate Computer Science (CS) programs [1][2]. While this surge of future programmers bodes well for a burgeoning industry [3], universities are having trouble providing adequate resources for increasingly large classes of CS students. In particular, schools are struggling to provide timely access to help in office hours [4][5][6].

Timely feedback is important for student learning [7][8], and universities are using a variety of solutions to combat long wait times. One solution of interest is to increase the number of teachers available to help students in office hours through the use of peer teachers. According to Smith et. al. [9], peer teachers are ”more senior students who are further advanced in the program of study” that can provide one-to-one instruction in an office hours setting. Because the number of students candidates for peer teachers increases as the course increases, peer teachers can scale as classes continue to grow.

Another solution is to provide students with alternative methods of feedback using automated systems. One type of automated feedback is an Intelligent Tutoring Systems (ITS), which provides students with immediate, adaptive feedback through a variety of techniques including hints, syntactic debugging, and navigational direction [10][11][12]. Another type of automated feedback popular with computer science courses is an autograder. An autograder takes a student solution as input, runs it against a suite of test cases, and returns feedback to the student about the correctness of their solution [13][14][15].

2.1 Research Question

In-person peer teaching office hours and automated mechanisms are both scalable strategies to provide timely feedback to aid student learning. In this paper, we examine the relationship between the two methods, specifically:

- Is there an association between autograder feedback style and demand for office hours?

We investigate the relationship between autograder feedback and peer teaching feedback in office hours through an analysis of a large dataset covering many semesters of many courses, as well as through a case study. Our results lead to recommendations on how to better leverage automated feedback while still providing timely access to peer teachers in office hours.

3 Related Work

Our work looks at the help-seeking behaviors of students in one-to-one peer teaching office hours. Several studies have examined factors that influence students’ decisions to seek or avoid help in a classroom setting, finding that students are influenced by their perceptions of the help-seeking experience as well as their own achievement goals. For example, students that perceived an unwilling, incompetent helper or a long wait to receive help were more likely to avoid help [16]. Furthermore, students with relative-ability goals or social-status goals were more likely to avoid seeking help due to the perceived threats to their self-worth. In contrast, students with task-focused achievement goals were more likely to seek help as they were able to view help-seeking as a method to further their mastery of the subject matter [17].

Feedback is an important part of the learning process [18], and providing effective feedback is especially important when demand for help is high. Hattie and Timperley created a model for effective feedback [19]. Using this model, Ott et. al. created a road map for giving effective feedback in a CS1 class [20]. Feedback may be given by humans or by automated feedback mechanisms.

Automated feedback is one technique to provide timely, scalable feedback. In their roadmap, Ott et. al. note that automated tools that deliver adaptive feedback can be used at the process level to assist students [20]. Several studies have implemented automated feedback systems to help students gain information about the syntactic and logical correctness of their solutions, the correctness and breadth of their test cases, and quality of the their code [21][22][14][23][15]. More advanced systems can provide students with targeted feedback by learning and classifying common problems [24][25].

Web-based office hours queues coupled with peer teaching is another technique for addressing feedback at scale. For example, the web-based queue by Smith et. al. has students request help from a peer teacher by adding themselves to a queue. Peer teachers help students, and the tool tracks interaction length and whether the student's question was answered. The tool was used in CS2 classes at three large research universities, and the study found that peer teaching office hours face many challenges including long wait times and an uneven distribution of instruction time. Because our data was collected using similar tools in a similar environment, we use the results of this study to validate our data. This paper then looks beyond the queueing tool to analyze how automated feedback mechanisms affect wait times.

4 Methods

In this section, we describe data collection from peer teaching office hours queues, the context of the computer science curriculum, the different types of automated feedback mechanisms, and our statistical methods.

The raw office hours queue data contains 195251 records, and after cleaning and filtering, there are 105941 records reflecting 17 unique courses: 2 100-level, 4 200-level, 2 300-level, and 9 400-level. The records occur between September 2016 and December 2019, before the COVID-19 pandemic began.

4.1 Data Collection

Our data set was collected from two web-based office hours queues used by the majority of classes at a large research university. Each course uses one of the two queues, and the core functionality of both is the same: a student joins a digital queue, and instructors remove the student after helping them. Both queues log timestamps when a student joins the queue and when an instructor removes the student after helping them. The queues are opened and closed when instructors hold office hours. If multiple office hours sessions are held during one day, the queue is cleared between sessions.

4.2 Context

Our data set is derived from the computer science curriculum at a large, public, research institution and spans 17 different courses, many with multiple semesters of data. All the courses in our study used an autograder and a digital office hours queue. In all cases, instructors held a consistent number of office hours each week. The curriculum begins with two options for the CS1 introductory programming classes, for engineering and non-engineering students. Then, students take CS2 and CS3, focusing on introductory data structures and programming in CS2 as well as algorithms and data structures in CS3. CS majors take a number of upper level courses of their choosing as well as hardware and theory. Computer science minors take a subset of the same courses.

4.3 Wait Time Calculation, Proxy for Demand

Students frequently wait during peer teaching office hours to get help with programming projects at various stages of completion [26]. Our web logs show timestamps when a student joins the office hours queue and when an instructor removes the student after helping them. To calculate wait time, we compute the difference between when a student joins the queue to when they are removed.

We use student wait time as a proxy for demand. The supply of office hours is fixed on a per capita basis. Instructors hold a fixed number of office hours each week, and this number is consistent throughout the semester. Later, we show that staff size increases linearly with student enrollment, indicating that classes large and small have comparable capacities to help students (Section 5.3).

4.4 Encounter Length Calculation

Encounter length is the time that a student spent with an instructor during office hours. We calculate encounter length from web log timestamps as the difference between two students removed from the queue by one staff member. For example, if an instructor helps two students in a row, x and y, the encounter length for student x is the $timestamp(y) - timestamp(x)$, where $timestamp(x)$ is the time student x was removed from the queue.

4.5 Cleaning and Filtering

We first merged the web logs, producing 195251 records. We removed 1657 entries due to lack of relevance. Specifically, entries for a class for doctoral students without candidacy status were not relevant to the study, and entries for special topics classes were all represented by the same course number, meaning that they were indistinguishable in the logs. In addition, we removed 12426 entries where a student removed themselves from the queue because they did not represent an encounter with a staff member.

Web logs are messy. For example, if an instructor forgot to clear the queue when office hours ended the previous day, our data set would contain an erroneous wait time of over 12 hours. Similarly, if an instructor was unable to find a student, they would remove that student from the queue, help the next student, and then remove the second student. Although the first student was not helped by a staff member, they would appear to have an encounter length of a few seconds. In addition, because of the way encounter length is calculated, it does not provide accurate numbers for the last student helped by a staff member for the day. We filtered out these erroneous records, resulting in 124454 remaining records.

Finally, we verified that only full semesters were part of the data set. We used enrollment data as well as the first and last encounter of the semester to determine if a semester was incomplete. After removing records corresponding to courses that lacked a full semester of web logs, our final data set contained 105941 records.

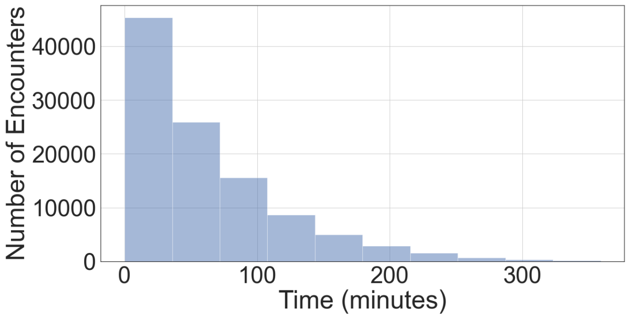


Figure 1: Distribution of wait times for peer teaching office hours. We use wait time to measure student demand. While the mean was 61 minutes, the long tail illustrates that many students experience frustratingly long wait times over 3 hours.

4.6 Summary Statistics

After merging, cleaning, and sanity checking the data, there are 105941 unique encounters made by 8258 unique students and helped by 424 unique instructors. The data is collected from 10 semesters ranging from Fall 2016 to Fall 2019. Figure 1 shows a distribution of wait times for all records, where the mean is 61 minutes, the median is 44 minutes, and the standard deviation is 58 minutes. It is clear from the long tail that student complaints have validity; many have waited over an hour to receive help from an instructor, and a nontrivial number of students have waited for 2 or 3 hours.

4.7 Autograder Feedback Mechanisms

Classes large and small use an autograder to provide feedback on programming assignments. When students upload their files to the autograder through a web user interface, their code is run against instructor-provided test cases. A student's grade is determined by the number and point value of the tests their code passes. Classes configure which tests to run, how much each test is worth, how often students are allowed to submit, and what type of feedback students receive. We note that none of the classes in our study used more advanced targeted feedback (e.g., [24][25]), but rather displayed varying levels of detail from test case output.

Some classes opt to show no feedback, so that students do not know if a test case succeeded or failed until after the project deadline has passed. In fact, students do not know which test cases exist until after the deadline. Other classes allow students to see pass/fail status but do not provide any additional information about why a test failed. Still other classes provide detailed stack trace or a partial output difference against the instructor solution in addition to pass/fail status. Classes may also choose to provide test case source code to students so that they are able to debug locally.

The predominate feedback type used by each class divided into four categories. The categorizations for each class in our data set are shown in Table 1.

- **Hidden code, no feedback** The test case source code run by the autograder is not available to students. Students receive feedback after the project deadline.

Course	Autograder Feedback Style
CS0 CS Pragmatics	Visible code, detailed feedback
CS1, Engineering	Hidden code, opaque feedback
CS1, Non-engineering	Hidden code, detailed feedback
CS2	Hidden code, no feedback
CS3	Hidden code, opaque feedback
Intro to Java	Hidden code, detailed feedback
Computer Organization	Hidden code, opaque feedback
Security	Hidden code, no feedback
Computer Vision	Hidden code, no feedback
Operating Systems	Hidden code, opaque feedback
Compilers	Hidden code, opaque feedback
Databases	Hidden code, opaque feedback
Web Systems (pre 2019)	Hidden code, opaque feedback
Web Systems (2019 onward)	Visible code, detailed feedback
Information Retrieval	Hidden code, no feedback
Programming Languages	Visible code, detailed feedback
Distributed Systems	Visible code, detailed feedback

Table 1: Classes and their automated feedback style.

- **Hidden code, opaque feedback** The test case source code run by the autograder is not available to students. Students see the pass/fail status of tests before the deadline.
- **Hidden code, detailed feedback** The test case source code run by the autograder is not available to students. Students see the pass/fail status, as well as a diff or traceback.
- **Visible code, detailed feedback** The test case source code run by the autograder is available for the students to debug locally on their own computers. The autograder shows students the pass/fail status of a test as well as debugging information.

4.8 Statistical Methods

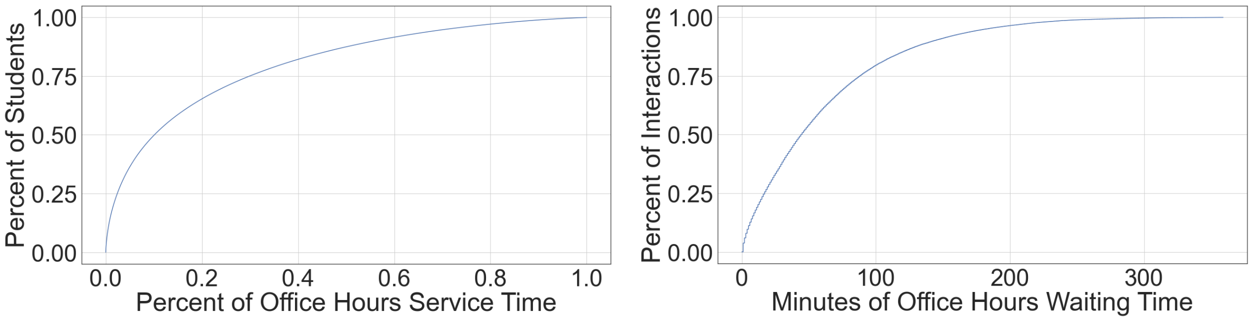
We use several statistical tests to analyze our data set. To compare the wait times for office hours in different classes using different autograder feedback styles, we use one-way analysis of variance (ANOVA). To compare the office hours wait times of a class before and after a new autograder feedback style, we use a two-sided, independent sample t-test.

5 Results

We now examine the relationship between automated feedback style and office hours wait time. We first compare our data set to previous work. Then, we rule out course enrollment and staff size as possible confounding variables. Using an ANOVA test, we analyze the association between office hours wait times and different autograder feedback styles.

5.1 Comparison with Prior Work

Our web-based queue for peer teaching office hours is similar to Smith et. al. [9]. The context is similar, with both being deployed in computer sciences courses at larger universities.



(a) Total service time per student

(b) Interaction wait times

Figure 2: CDF of total peer teaching office hours time used by each student, and wait times. The X-axis is ordered from smallest to largest. These results closely resemble the results of [9], showing that our dataset is comparable to other institutions.

Figure 2a shows a cumulative distribution function (CDF) of total peer teaching office hours time used by each student over the course of the semester. Like Smith, we see that a small percentage – around 10% – of students accounts for nearly half of all office hours. Figure 2b shows a CDF of wait times over the semester. Like Smith, around 40% of wait times exceeded one hour.

5.2 Enrollment and Wait Time

First, we ruled out total course enrollment as a possible confounding variable. Figure 3a shows median wait time vs. course enrollment. Each dot represents one semester of one course, and dots of the same color represent the same course. The 60th, 70th, 80th, 90th, and 95th percentiles displayed similar trends.

We observed near-zero correlation between median office hours wait times and course enrollment. A linear regression resulted in an R-squared value was 0.03. Even the 95th percentile wait time only had an R-squared value of 0.17.

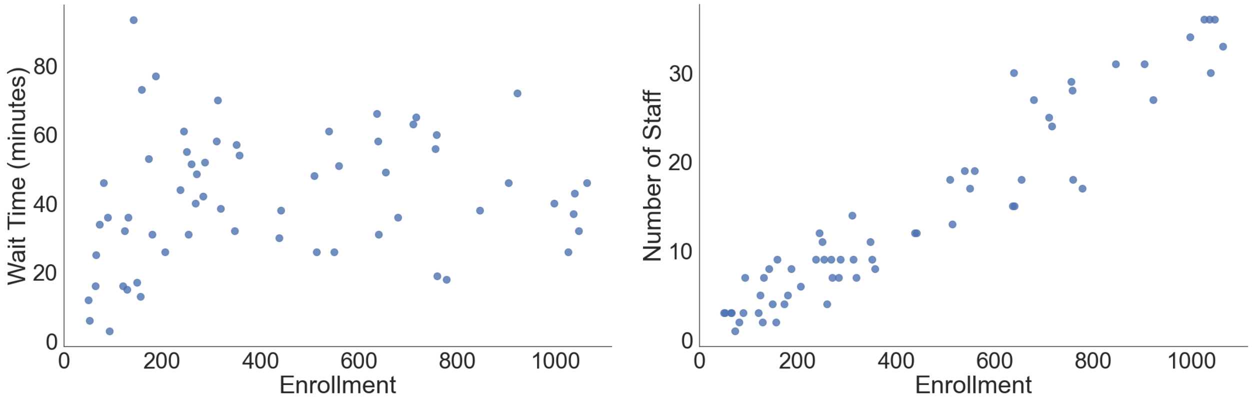
5.3 Staff Size and Wait Time

Next, we ruled out staff size (number of teaching assistants) as a possible confounding variable. We calculated staff size as the number of unique TAs in one semester. A visual representation of enrollment and staff size is shown in Figure 3b. As in Figure 3a, each dot represents a single class for a single semester.

The number of TAs is strongly, positively correlated with the class's enrollment, with an R-squared value of 0.97. Thus, the student-staff ratio remains consistent as class size increases, suggesting that classes have proportional capacity to help students whether they are large or small.

5.4 Autograder Feedback Style and Wait Time

We show the average encounter length and wait time for peer teaching office hours disaggregated by autograder feedback style in Figure 4. Descriptions of the feedback styles are in Section 4.7.



(a) Median wait time vs. course enrollment

(b) Staff size (TAs) vs. course enrollment

Figure 3: Ruling out confounding variables enrollment and staff size. Each dot represents one class during one semester. There was no statistically significant association between median wait time and enrollment, thus we ruled it out as a confounding variable. Staff size and enrollment were strongly, positively correlated. We ruled out staff size as a confounding variable after seeing that larger classes have the same capacity to help students in peer teaching office hours as small classes.

	df	Sum Sq.	Mean Sq.	F	PR(>F)
Autograder feedback style*	3.0	7670.133	2556.711	6.951	4.5e-4
Residual	58.0	21334.764	367.841		

Table 2: Results of ANOVA test where the independent variable was autograder feedback style and the dependent variable was office hours wait time. *Statistically significant.

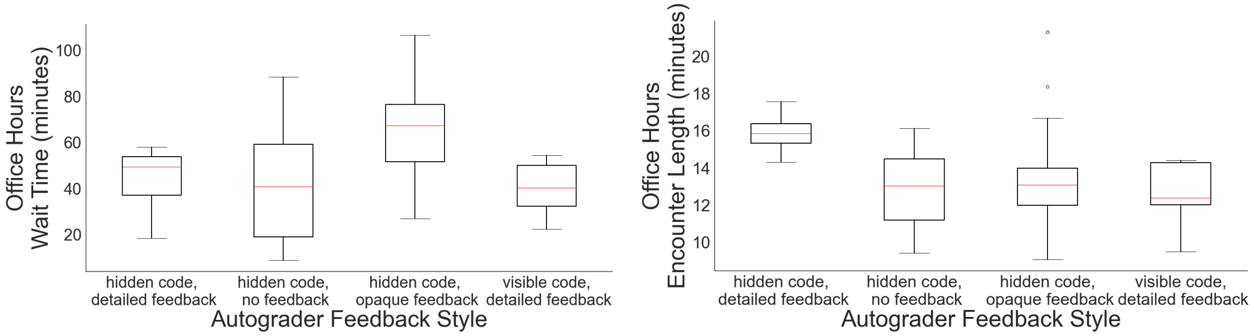
The encounter length magnitudes for all feedback types were very similar, with ”Hidden Code, Detailed Feedback” having the largest mean encounter length of 16 minutes. The other feedback styles all had mean encounter lengths of 13 minutes. When examining wait time, the ”Hidden code, opaque feedback” feedback style stood out with a mean wait time of 63 minutes. The other feedback styles had mean wait times of 40, 41 and 44 minutes.

We computed a one-way analysis of variance (ANOVA) where the independent variable was autograder feedback style, a 4-valued discrete variable, and the dependent variable was the wait time for peer teaching office hours, a continuous variable (Table 2). We observed a statistically significant association between feedback style and wait time ($p = 0.00045$).

6 Case Study: Course Intervention

This case study examines the demand for peer teaching office hours in one course before and after changing the automated feedback mechanism. Earlier, we observed that ”Hidden code, opaque feedback” was associated with much higher wait times (Section 5.4). What happens when a course changes its automated feedback mechanism from the ”Hidden code, opaque feedback” to ”Visible code, detail feedback”?

This between-subjects experiment design used two semesters of the same course. The first semester was the control group and used ”Hidden code, opaque feedback”. The following



(a) Wait time by autograder feedback style

(b) Encounter length by autograder feedback style

Figure 4: Average wait time for office hours in courses with opaque autograder feedback was higher. While ”hidden code, detailed feedback” shows higher encounter lengths, the magnitudes were similar across feedback types.

semester was the experiment group which used ”Visible code, detailed feedback”.

6.1 Context

The context is Web Systems, an upper level class, during two consecutive course offerings. The class covers web applications, networking protocols, web security, scalability, and web search. In both semesters, the class was taught by the same professor with the same material, the same projects and the same project test cases. We note that some teaching assistants changed, but they were hired with the same criteria and in the same proportion to the number of students. The class consists of five projects and two exams, all graded on correctness.

6.2 Control and Experiment Groups

In the control group (n=357), the class used an autograder with the ”Hidden code, opaque feedback” style. The autograder displayed the pass/fail status of all test cases, but students did not have access to the test case source code.

In the experiment group (n=441), the class used an autograder with the ”Visible code, detailed feedback” style. Two thirds of the tests were public, where the test case source code was published to students. When a student failed a public test case, the autograder displayed a detailed diff between the correct output and the student output as well as a stack trace. One third of the test cases were withheld until after the deadline. Thus, before the deadline, students had full source code for all tests visible on the autograder.

6.3 Analysis

In the control group, the average office hours wait time was 69 minutes (n=357), while the average wait time in the experiment group was 52 minutes (n=441). A two-sided, independent sample t-test showed that the 24% decrease in the mean wait time was statistically significant, $p = 6.258\text{e}{-15}$.

7 Discussion

Our results show that autograder feedback style was associated with student demand for peer teaching office hours. While automated feedback is intended to improve student access to timely feedback, our results suggest that it may have a counter-intuitive side effect when the feedback is opaque, resulting in a surge of demand for peer teaching office hours.

7.1 Corroborating Prior Work

Our results corroborate the findings of Smith et al. [9]. Both examined computer science courses at large research universities and use similar web-based office hours queues. Our data set contains 105941 records collected from 17 unique courses over more than 3 years at one university, while Smith et al. collected approximately 3720 records from 3 unique courses during 2 semesters at 3 universities. As seen in Figure 2a, our results corroborate the finding that a small percentage of students occupy 50% of all available office hours. Furthermore, Figure 2b shows that our wait time results mirror Smith et al.'s in that around 40% of interactions required students to wait for at least one hour.

7.2 Larger classes do not have longer wait times

We ruled out enrollment and staff size as variables that might influence office hours wait time. All classes had a similar TA-to-student ratio and all TAs in all classes held a fixed number of office hours per week. Despite having the same per capita supply of office hours, both small and large class sizes exhibited both long and short wait times (Figure 3a). This observation contradicts the idea that only larger classes face problems with long wait times in office hours.

7.3 Opaque feedback associated with long waits

Classes whose autograders provided opaque feedback had substantially longer wait times compared to classes using other feedback styles (Section 5.4). The ”Hidden code, opaque feedback” autograder style had a 43 - 57% longer wait time than other autograder policies.

When a student fails a test case, they see the failed status on the autograder, but have little information to help them narrow down the root cause of the problem. While some students may be able to find the problem through their own independent testing, our results suggest that this kind of feedback may drive many students to office hours for more information about the failure.

Several classes that use autograders with hidden code, opaque feedback provide students with feedback in a different way, using student-written test cases. During grading, the autograder runs student-written test cases against the instructor solution to verify the validity of the student-submitted tests. Then, it runs student test cases on the student-submitted student code and reports any failures. In this way, students are able to create their own test suite to test their code against. Unfortunately, this method does not seem to curb the demand for office hours.

7.4 Encounter length is not the problem

One possible explanation for longer wait times is that students are being helped for a longer period of time, suggesting that classes with opaque feedback would have longer encounter lengths. However, Figure 4b contradicts this theory. Classes using opaque feedback did not have substantially longer encounter lengths compared to classes using other feedback styles. These findings suggest that there is a greater demand for office hours in classes with opaque automated feedback.

7.5 Changing feedback improves wait times

In our case study, we found that there was a statistically significant decrease in mean office hours wait time when a web systems class switched from using a ”Hidden code, opaque feedback” autograder policy to a ”Visible code, detailed feedback” autograder policy.

Opaque feedback may incentivize students to come to office hours with questions about why a test is failing, rather than questions about course concepts. In the control group, students received opaque autograder feedback, where students know that their code is failing, but have very little information about why. Anecdotally, the most frequent question in office hours was, ”Why am I failing this test case?” In addition, because students in the control group were able to see the pass/fail status of all the test cases being run, they may have been using autograder to test their code rather than independently creating their own test suite. Furthermore, because students were able to see their current grade, they could see whether or not they had achieved a perfect score, and could continue to come to office hours until they had received 100%.

Detailed feedback may drive students to come to office hours with questions about course concepts and use cases of the code rather than questions about test case failures. In the experiment group, the autograder provided detailed feedback to students when they failed a test case and test case source code was publicly available for students to run locally. As a result, students did not need to go to office hours to diagnose why they were failing an opaque test case. We leave the investigation of student questions related to the autograder to future work.

7.6 Some automated feedback types have counter-intuitive side effects

The literature is clear that feedback should be timely [8][7], and in general, automated feedback mechanisms like autograders can be used to provide timely feedback at scale. However, our results suggest that some automated feedback may counter-intuitively increase the amount of time it takes for students to get useful feedback from peer teachers. As a result, it is important that instructors carefully consider how the automated feedback they provide impacts student help-seeking behaviors.

8 Limitations

In this Section, we discuss threats to validity including slow office hours days, lack of information about students that never received help, and missing question context.

We calculate the length of an office hours encounter by looking at the difference between when a student was removed from the queue and when the next person was removed from the queue. In the common case, this is an accurate method, however, a slow day in office hours could lead to large breaks between students signing up to receive help. This could be fixed in future studies by also logging when a student is finished being helped.

Another limitation of our data set is a lack of information about students who never received help. At the end of a day, courses typically clear the office hours queue, which could still have students who did not receive help that day. The number of students left on the queue at the end of the day would provide an additional metric to describe student pain points in office hours. However, our data set does not contain this information.

Finally, our data set does not include information about what types of questions were being asked by students. As a result, all interactions, regardless of if they are related to a coding assignment, are included in our data set. Anecdotally, because wait times are currently very long, most students that come to office hours ask questions about coding assignments and ask admin or content questions on the class' online forum. Furthermore, peak office hours usage (and in turn, wait times), come just before project deadlines. However, future work could be done to categorize interactions by question type and filter the data further.

9 Conclusions and Future Work

This study examined the relationship between autograder feedback style and demand for peer teaching office hours in a large computer science program. We found that there was a statistically significant association between feedback style and wait times for peer teaching office hours, with the ”Hidden code, opaque feedback” style having 43-57% longer wait times than other feedback styles. Upon further investigation, we observed a 24% decrease in mean wait time when a class switched from opaque feedback to detailed feedback with published test case source code. These results suggest that some techniques intended to provide timely automated feedback counter-intuitively increase the time students wait for in-person feedback.

Future work may improve our understanding of how autograder feedback styles affect student help-seeking behavior. First, work could be done to understand the relationship between help-seeking behavior and variables such as student demographics and grades. Peer teacher training could be studied together with the feedback they provide. Work could be done to understand how automated feedback style affects student perceptions of themselves, their work, and the class, and if these perceptions affect their likelihood of seeking help. Finally, categorization and analysis of student questions could be used to improve automated feedback.

By considering the relationship between automated feedback mechanisms and demand for in-person feedback, instructors can best utilize automated feedback mechanisms while ensuring that students get timely access to help in office hours.

References

[1] A. Baer and A. DeOrio, “A longitudinal view of gender balance in a large computer science program,” in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 23–29. [Online]. Available: https://doi.org/10.1145/3328778.3366806

[2] E. National Academies of Sciences and Medicine, *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments*. Washington, DC: The National Academies Press, 2018. [Online]. Available: https://www.nap.edu/catalog/24926/assessing-and-responding-to-the-growth-of-computer-science-undergraduate-enrollments

[3] U.S. Bureau of Labor Statistics. (2020, Apr) Computer and information technology occupations : Occupational outlook handbook. [Online]. Available: https://www.bls.gov/ooh/computer-and-information-technology/home.htm

[4] K. Basu, “Some cis courses are so overloaded that students wait more than an hour for homework help,” Dec 2018. [Online]. Available: https://www.thedp.com/article/2018/12/cis-120-office-hours-wait-time-penn-upenn-philadelphia

[5] L. Weinstein, “Students, professors discuss finding resources for extra help in university’s largest classes,” Feb 2019. [Online]. Available: https://www.michigandaily.com/section/academics/students-professors-discuss-finding-resources-extra-help-university’s-largest

[6] J. Liu, “Cs in crisis: Is stanford doing enough to respond to capacity and inclusion challenges?” Feb 2019. [Online]. Available: https://www.stanforddaily.com/2019/02/19/cs-in-crisis-is-stanford-doing-enough-to-respond-to-capacity-and-inclusion-challenges/

[7] R. Higgins, P. Hartley, and A. Skelton, “The conscientious consumer: Reconsidering the role of assessment feedback in student learning,” *Studies in Higher Education*, vol. 27, no. 1, pp. 53–64, 2002. [Online]. Available: https://doi.org/10.1080/03075070120099368

[8] P. Ramsden, *Learning to teach in higher education*, 2nd ed. Routledge, 2003.

[9] A. J. Smith, K. E. Boyer, J. Forbes, S. Heckman, and K. Mayer-Patel, “My digital hand: A tool for scaling up one-to-one peer teaching in support of computer science learning,” in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 549–554. [Online]. Available: https://doi.org/10.1145/3017680.3017800

[10] T. Crow, A. Luxton-Reilly, and B. Wuensche, “Intelligent tutoring systems for programming education: A systematic review,” in *Proceedings of the 20th Australasian Computing Education Conference*, ser. ACE ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 53–62. [Online]. Available: https://doi.org/10.1145/3160489.3160492

[11] J. Yoo, C. Pettey, S. Yoo, J. Hankins, C. Li, and S. Seo, “Intelligent tutoring system for cs-i and ii laboratory,” in *Proceedings of the 44th Annual Southeast Regional Conference*, ser. ACM-SE 44. New York, NY, USA: Association for Computing Machinery, 2006, p. 146–151. [Online]. Available: https://doi.org/10.1145/1185448.1185482

[12] J. C. Nesbit, O. O. Adesope, Q. Liu, and W. Ma, “How effective are intelligent tutoring systems in computer science education?” in *2014 IEEE 14th International Conference on Advanced Learning Technologies*, 2014, pp. 99–103.

[13] M. Ball, “Lambda: An autograder for snap!” Master’s thesis, EECS Department, University of California, Berkeley, Jan 2018. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2018/EECS-2018-2.html

[14] J. Gao, B. Pang, and S. S. Lumetta, “Automated feedback framework for introductory programming courses,” in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, ser.

ITiCSE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 53–58. [Online]. Available: https://doi.org/10.1145/2899415.2899440

[15] R. Sharrock, P. Bonfert-Taylor, M. Hiron, M. Blockelet, C. Miller, M. Goudzwaard, and E. Hamonic, "Teaching c programming interactively at scale using taskgrader: An open-source autograder tool," in *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*, ser. L@S '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3330430.3333670

[16] C. Midgley, A. Ryan, and P. Pintrich, "Avoiding seeking help in the classroom: Who and why?" *Educational Psychology Review*, vol. 13, 06 2001.

[17] A. Ryan and P. Pintrich, ""should i ask for help?" the role of motivation and attitudes in adolescents' help seeking in math class," *Journal of Educational Psychology*, vol. 2, pp. 326–341, 01 1997.

[18] V. J. Shute, "Focus on formative feedback," *Review of Educational Research*, vol. 78, no. 1, pp. 153–189, 2008. [Online]. Available: https://doi.org/10.3102/0034654307313795

[19] J. Hattie and H. Timperley, "The power of feedback," *Review of Educational Research*, vol. 77, no. 1, pp. 81–112, 2007. [Online]. Available: https://doi.org/10.3102/003465430298487

[20] C. Ott, A. Robins, and K. Shephard, "Translating principles of effective feedback for students into the cs1 context," *ACM Trans. Comput. Educ.*, vol. 16, no. 1, Jan. 2016. [Online]. Available: https://doi.org/10.1145/2737596

[21] J. Perretta and A. DeOrio, "Teaching software testing with automated feedback," *2018 ASEE Annual Conference amp; Exposition Proceedings*, 2018. [Online]. Available: https://www.asee.org/public/conferences/106/papers/21636/view

[22] H. Keuning, J. Jeuring, and B. Heeren, "A systematic literature review of automated feedback generation for programming exercises," *ACM Trans. Comput. Educ.*, vol. 19, no. 1, Sep. 2018. [Online]. Available: https://doi.org/10.1145/3231711

[23] R. Singh, S. Gulwani, and A. Solar-Lezama, "Automated feedback generation for introductory programming assignments," *SIGPLAN Not.*, vol. 48, no. 6, p. 15–26, Jun. 2013. [Online]. Available: https://doi.org/10.1145/2499370.2462195

[24] A. Head, E. Glassman, G. Soares, R. Suzuki, L. Figueredo, L. D'Antoni, and B. Hartmann, "Writing reusable code feedback at scale with mixed-initiative program synthesis," in *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, ser. L@S '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 89–98. [Online]. Available: https://doi.org/10.1145/3051457.3051467

[25] G. Haldeman, A. Tjang, M. Babeş-Vroman, S. Bartos, J. Shah, D. Yucht, and T. D. Nguyen, "Providing meaningful feedback for autograding of programming assignments," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 278–283. [Online]. Available: https://doi.org/10.1145/3159450.3159502

[26] Y. Ren, S. Krishnamurthi, and K. Fisler, "What help do students seek in ta office hours?" in *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ser. ICER '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 41–49. [Online]. Available: https://doi.org/10.1145/3291279.3339418