Delivering Round-the-Clock Help to Software Engineering Students Using Discord: An Experience Report

Kathryn Bridson
University of Memphis
Memphis, Tennessee, USA
kbridson@memphis.edu

Jeffrey Atkinson
University of Memphis
Memphis, Tennessee, USA
Jeff.Atkinson@memphis.edu

Scott D. Fleming
University of Memphis
Memphis, Tennessee, USA
Scott.Fleming@memphis.edu

ABSTRACT

This experience report describes the delivery of round-the-clock help to students using Discord (a popular messaging and voice/video calling platform) in a remote software engineering course. Students in the course learn full-stack web development using Ruby on Rails and PostgreSQL, and work in teams to develop web applications. Our central goal in offering round-the-clock help using Discord was to increase the amount of help that students receive from teachers (i.e., teaching assistants and the instructor). Indeed, we found that our 24/7-Discord approach led to a considerable increase in the amount of student–teacher interaction versus the approach used previously, which emphasized in-person office hours and a question-and-answer forum in Piazza. Moreover, students from underrepresented groups in computer science interacted with teachers at a rate comparable to other students, and we received consistently positive feedback from students regarding the approach. We also made several key observations about when students tended to seek help, including that they sought help the most between 7:00 p.m. and midnight, that help seeking spiked right before deadlines, that students posted the fewest help messages on weekends, and that students posted significantly more messages during the first half of the course, which emphasized skills assignments, versus the second half, which focused on team project work.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

software engineering education, help seeking, round-the-clock help, Discord, Piazza, teaching assistants, full-stack web development

ACM Reference Format:
Kathryn Bridson, Jeffrey Atkinson, and Scott D. Fleming. 2022. Delivering Round-the-Clock Help to Software Engineering Students Using Discord: An Experience Report. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499385

1 INTRODUCTION

This experience report describes the use of Discord [29] (a popular messaging and voice/video calling platform) to deliver round-the-clock help to students in a remote undergraduate software engineering (SE) course. The course follows an approach used by other SE courses in the literature (e.g., [7, 10]), teaching SE principles in the context of "full-stack" web development—that is, the development of both client and server software built upon a platform that includes a web server framework and a database management system. Students spend the first half of the course learning how to develop full-stack web applications using Ruby on Rails and PostgreSQL, and during the second half, they work in collaborative teams to build Rails-based web applications.

Help from a teacher (i.e., a TA or instructor) can benefit students in the course considerably. Throughout both halves of the course, there are numerous technical challenges that students may encounter, such as difficulties maintaining a working development environment, subtle bugs in their code, uncertainty about how to use unfamiliar code libraries and APIs, and questions about how to implement desired functionality. Often, the most efficient and most educationally beneficial way to address these challenges is to get help from a teacher. For example, the teacher may not only help the student solve their problem, but they may also provide additional helpful feedback and coaching (similar to that found in cognitive apprenticeship [5]). Teachers may also benefit from such help interactions with students, for example, by adding to their pedagogical content knowledge or by inspiring improvements to their instructional materials.

Despite the potential benefits of teacher help, the amount of help that students have tended to receive in the course has always seemed lacking. Prior iterations of the course were held in person, and students were offered two main options for receiving help. One option was to come to the teachers' office hours, which were typically held immediately following class or by special appointment. The other option was to use the question-and-answer system, Piazza [31], to post questions to an online help forum. Based on our prior experiences, only a few students would ever come to office hours, and based on Piazza's usage statistics, fewer than half of the students ever posted a message in Piazza.

Reflecting on this under-utilization of teacher help, we identified two key barriers that may be to blame. One barrier pertains to the accessibility of in-person help. Although we carefully schedule the regular office hours so as not to conflict with any other CS courses, those times may still be undesirable to students because, for example, they may have some other scheduling conflict, or the times that they need help may not coincide with when the office hours are offered. Another barrier pertains to the communication

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499385

features and response time associated with Piazza. Piazza offers only text messaging (no voice or video), and some problems may be difficult to diagnose and explain via text only. Moreover, although the teachers always make an effort to respond to student posts in a timely fashion, there are certain times of the day when they are typically unavailable, such as late in the evening and at night.

To address these barriers, we tried a new approach for delivering help with two key aspects: (1) the use of Discord for communication with students and (2) 24/7 availability of a teacher to respond to help requests. We chose Discord, because it offers a rich set of features for various modes of interaction with students. It has text-forum features that can be used for asynchronous as well as synchronous texting. Unlike Piazza, it also has features for voice/video communication and for screen sharing. We particularly valued the addition of these features, because we aimed to use Discord as a replacement for in-person office hours, which were disallowed due to the COVID-19 pandemic. It is fair to note that other platforms exist with features similar to those of Discord (e.g., Slack [32] and Campuswire [4]); however, we went with Discord, because it met our needs, we were familiar with how to use it, and we were aware that many undergraduate students were already Discord users. To provide 24/7 availability, we added two additional TAs whose main responsibility was to monitor Discord and respond to help requests. These TAs arranged their schedules to cover all hours of the day and all days of the week.

In applying our 24/7-Discord approach to help delivery, we sought to achieve two key goals:

(1) to increase the *amount of help* that students receive from teachers and
(2) to better understand *when* students tend to seek help.

Additionally, understanding the effectiveness of our approach for students from *underrepresented groups in CS* [17] was a key concern. Such students often encounter barriers that are unexpected to even well-meaning teachers (e.g., [28]), so it is important to evaluate educational interventions for their impacts on those students.

2 BACKGROUND AND RELATED WORK

2.1 Help-Seeking Behaviors of CS Students

Our primary interest in this work is helping students with coding problems, and studies have shown that such technical help is indeed the type that CS students seek most. For example, studies have reported this trend in introductory CS courses [21, 26] as well as in a web development course [19].

Prior studies have also revealed trends in when CS students seek help. Numerous studies have found a tendency among CS students to procrastinate on their work [1, 11, 12, 18, 22, 23], thus leading to spikes in help seeking right before deadlines [21]. Regarding the times of day that students will tend to seek help, one study [33] investigated when CS1 students worked, and identified four chronotypes (patterns of when one tends to be active during the day). The "evening" chronotype (active 8:00 p.m. to midnight) was least common among U.S. students (15%), whereas the "napper" chronotype (active in the morning and from late afternoon to late evening) was most common (40%). Moreover, only 1% of activity occurred between midnight and 5:00 a.m. Chronotype was also a predictor of success, with evening students tending to have lower exam grades.

Prior work has also identified the accessibility of help as an important concern. For example, in one study comparing help from a human tutor versus an intelligent tutoring system (ITS), students reported that accessibility was a key positive aspect of the ITS [20]. Improving students' access to teachers was also a key consideration of Malan's virtualized office hours [15]; however, he used the now-deprecated Elluminate system [30] for communication, and reported persistent issues with long wait times to receive help.

2.2 Discord and Slack in CS Education

Discord is a communication platform that includes features for text messaging and voice/video calling [29]. In Discord, a *server* is a virtual space that can be created for a community of users (in our case, the teachers and students affiliated with a course). A server can contain *text channels* and *voice channels*. A text channel enables users to communicate by posting text messages, and a voice channel enables users to communicate via video calling and screen sharing. To facilitate finding and navigating channels within a server, they can be organized into *categories*. Discord provides a wide range of access control options—for example, membership to a server can be open to anyone or can be by invitation only, and access to categories and channels can be restricted to specified users or groups of users.

Because Discord is relatively new (originally released in 2015, with video calling and screen sharing added in 2017), there have been only a few works to date that report on its use in CS education. Others have reported using Discord for a variety of purposes, including for community building [16, 27], as a virtual classroom [6, 27], for student collaboration and teamwork [13, 14], and for online tutoring [16]. However, only a few results have been reported that are relevant to our goal of delivering help to students. In one study [14], 68% of students in a CS course reported that Discord helped team members in assisting each other; however, the teamwork did not involve coding, and help from instructors or TAs was not mentioned. Another study on the use of Discord for tutoring and community building reported high utilization of Discord among students and positive feedback from students on Discord [16]; however, the extent to which these results were applicable to the delivery of teacher help was not clear.

Slack [32] is a system with text-messaging features similar to Discord's that has also been used in CS education. In one particularly relevant study [25], Slack was used for communication among students and instructors in a software architecture course. The study found that 79% of students in the course reported that Slack helped them get help from teachers or other students quickly. This result bodes well for our approach, which further adds 24/7 availability of expert helpers, unlike the prior work.

2.3 Piazza in CS Education

Piazza, the system used previously in our course to provide online help, has also been widely used to deliver help to students in CS education. Although Piazza lacks video chat and screen sharing, it does provide a question-and-answer (Q&A) forum, with many features that overlap Discord's text messaging features. The main differences between the messaging functionality provided by Piazza versus Discord might best be characterized by the differences in their high-level user interaction designs. Discord's text messaging features aim mainly to support informal discussions, whereas Piazza's Q&A features are a bit more formal, aiming to get users to produce high-quality reusable documentation that includes a clear and detailed question along with one or more equally clear and detailed answers. This tension between informality of user interaction versus production of reusable documentation is perhaps the key differentiator between the features of Discord and Piazza.

Perhaps because of Piazza's slightly older age (first wide release was in 2011) and its focus on education, more has been studied and reported about its uses in CS education than has been for Discord. Piazza has been described as a invaluable tool for CS educators [9], although some have mentioned issues with getting students to use it [8]. Engaging underrepresented students has been reported as a benefit of Piazza, with one study [24] finding that women post more questions on Piazza than do men. The study also noted the potential importance of being able to post anonymously, as women made greater use of that option than did men. Although anonymous posting was available in our prior deployments of Piazza, it was not an option in our Discord deployment. Thus, we thought it important to check for any potential issues with how students from underrepresented groups engage with Discord. The level of student engagement in Piazza has also been found to be important, with two studies [2, 26] finding that greater engagement in Piazza tended to predict higher grades. Unfortunately, in our prior experiences with Piazza, student engagement tended to be low, and thus, we were motivated to try our new 24/7-Discord approach in the hopes of increasing engagement.

3 24/7-DISCORD APPROACH

3.1 Discord Configuration

To deliver help to students, we configured our Discord server with several different channels to provide different types of help. For technical help (e.g., coding-related bugs and questions), we provided a text channel, `#help-help-help`. This channel was by far the most used by students seeking help, and it was generally the first place that students would go for help. For non-technical help (e.g., questions about course policies and instructions), we provided another text channel, `#questions-non-technical`. Although our intent was keep the different types of questions separated using these channels, students posted a number of technical questions to the `#questions-non-technical` channel anyway.

For help interactions using video calling and screen sharing, we created three "help rooms." Figure 1 illustrates how each help room was configured. The voice channel was primarily used as a way for the teacher to talk to the student via telephony and, if necessary, for the student to share their screen with the teacher. The help room's text channel was primarily used as a sort of virtual whiteboard, enabling the student and teacher to conveniently share text and images.

In addition to delivering help, we also configured Discord to support other aspects of the course. There were several specialized text channels: `#announcements` for the instructor to broadcast announcements to the class, `#feedback-on-the-course` for gathering and discussing students' feedback, and `#general` for miscellaneous comments and discussion. There were also a number of special-purpose "rooms," each with a text channel and a voice channel, similar to the help rooms. A "lecture hall" room was used to hold course lectures. Each student team had their own "team room" that they used to hold meetings and work collaboratively. Finally, there was an "instructors-only room" that was visible only to the instructor and teaching assistants.

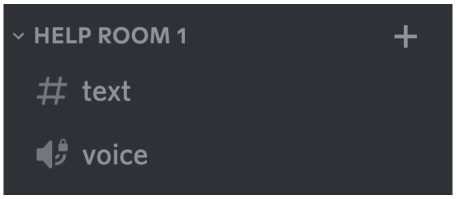

Figure 1: Example help room, each of which contained a text channel (`#text`) and a voice channel (`voice`), which supported video calling and screen sharing.

3.2 Teachers' Round-the-Clock Schedule

Three teachers affiliated with the course provided help to students using Discord (1 female, all White). The first and second authors were teaching assistants and were responsible for providing most of the help to students in Discord. The third author was the course instructor and also provided some help to students in Discord.

The two teaching assistants arranged their schedules, so at least one of them would be available at any given time during the week. In particular, one of them covered late morning through evening (roughly 8:00 a.m. to 10:00 p.m.), and the other covered late afternoon through early morning (roughly 3:00 p.m. to 8:00 a.m.). During these periods, each of them kept watch for Discord notifications, and they made it a point to respond to any student messages as quickly as possible (often immediately and generally in no more than 10–15 minutes). Additionally, the course instructor would sporadically provide help to students, mostly on weekdays during working hours.

4 COURSE EXPERIENCES

4.1 Course Context

The software engineering course lasted roughly 14 weeks. The first 7 weeks emphasized training the students in full-stack web development using Ruby on Rails and PostgreSQL. Each week, student were tasked with completing a *skills assignment* that involved three parts. First, the students performed an *active reading* of a set of provided worked examples, which involved not only reading the worked examples, but performing the steps on their own computers. Second, the students completed a low-stakes *practice test* in which they were presented a task that was essentially isomorphic to tasks covered in the worked examples. Third, the students recorded an *explanation video* in which they performed yet another isomorphic task while explaining what they were doing and why as they performed each step. The last 7 weeks of the course involved a collaborative team project in which the students worked in 3- to 4-person teams to develop a web app for a customer (a role played by another student in the course). The first 3 weeks of the project were devoted to initial planning, design, and setup and was followed by two 2-week

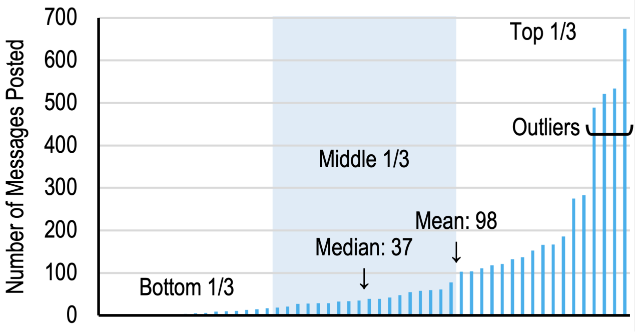

Figure 2: Number of Discord posts made by each student.

development iterations in which students did the bulk of the coding. At the end of each development iteration, each team had to (among other things) record a video demonstrating their working app. It was required that, during each development iteration, each and every member of a team made substantive coding contributions to the project.

There were 52 students enrolled in the course. Regarding gender, 14 students (27%) identified as female, and the rest identified as male. Following the NSF's definition [17], 22 students (42%) belonged to an underrepresented race and ethnicity in computer science (6 Hispanic or Latino, 11 Black or African American, and 5 mixed).

4.2 Usage Analysis Method

To assess the extent to which students received help, we analyzed the frequency with which they posted messages. In the case of Discord, we analyzed a log of message-posting events. Piazza, on the other hand, provided summary statistics regarding the messages students posted. Unfortunately, we were unable to collect data regarding students' voice-channel usage; however, we will discuss the teachers' anecdotal observations in Section 5.

4.3 Descriptive Statistics of Usage

Students in the course posted a total of 5105 messages in Discord. Figure 2 shows the number of messages that each student posted. The top third of students posted between 103 and 674 messages. The middle third posted between 21 and 78 messages. The bottom third posted 19 or fewer messages. The four highest message counts (489, 521, 534, and 674) were noticeably greater than the others and were considered statistical outliers.

The three teachers posted a total of 2890 messages in Discord. The teaching assistants posted 1466 and 1233 messages, respectively, and the course instructor posted 191 messages. For sake of space, we won't report further data on the teachers; however, their activity data (e.g., when they posted messages) is strongly correlated with the students' data, as would be expected.

4.4 Comparison with Prior Semesters

As shown in Figure 3, students posted considerably more help messages during the semester in which round-the-clock help via Discord was available versus prior semesters in which help was offered during more limited hours via Piazza. Indeed, the difference between the 24/7-Discord group and each of the Not-24/7–Piazza groups was statistically significant, with a t-test reporting $p < 0.0001$ for each comparison. The differences between the 24/7-Discord semester and the Not-24/7–Piazza semesters are particularly striking when we consider the number of students who posted any messages at all. During the 24/7-Discord semester, nearly every student in the course posted at least one message (47 out of 52, 90%), whereas during the each of the Not-24/7–Piazza semesters, less than half of the students ever posted a message (F19: 13 out of 35, 37%; S19: 11 out of 27, 41%; F18: 20 out of 52, 38%;).

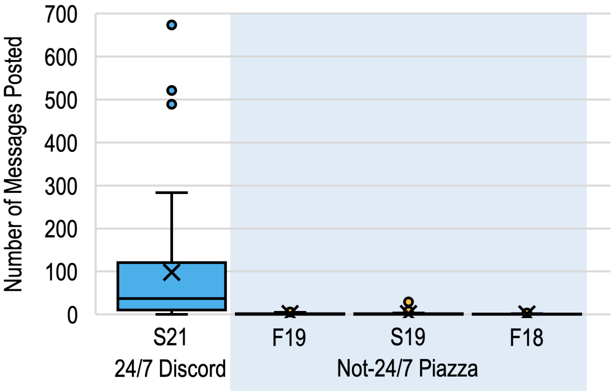

Figure 3: Number of messages students posted using Discord in Spring 2021 versus Piazza in prior semesters.
24/7 Discord, Spring 2021: $N = 52$, $M = 98.2$, $SD = 150.3$
Not-24/7 Piazza, Fall 2019: $N = 35$, $M = 1.5$, $SD = 3.1$
Not-24/7 Piazza, Spring 2019: $N = 27$, $M = 2.0$, $SD = 5.7$
Not-24/7 Piazza, Fall 2018: $N = 52$, $M = 1.0$, $SD = 1.8$

4.5 Usage by Underrepresented Groups

As Figure 4 shows, there was little difference in the number of messages that students who belonged to underrepresented groups in computer science posted in Discord versus the number posted by other students. Indeed, no statistical difference was detected between any underrepresented group and the rest of the class (for each t-test, $p \geq 0.2$).

4.6 When Students Sought Help

4.6.1 *Times of Day.* Figure 5 shows when during the day students sought help. The peak times were between 7:00 p.m. and midnight, which account for nearly half of all messages posted (47%). Another significant period of activity was between midnight and 5:00 a.m., which accounted for nearly one-fifth of all messages (18%) Morning (5:00 a.m. to noon) was the least-active part of the day, accounting for less than one-tenth of all messages (8%). The course met on Tuesdays and Thursdays from 2:40 p.m. to 4:05 p.m., which may explain the uptick in messages between 3:00 p.m. and 5:00 p.m.

4.6.2 *Days of the Week.* Figure 6 shows how much help students sought on each day of the week. As the figure shows students posted the most messages on Tuesdays, Wednesdays, and Thursdays. Those three days accounted for over 70% of all the messages students

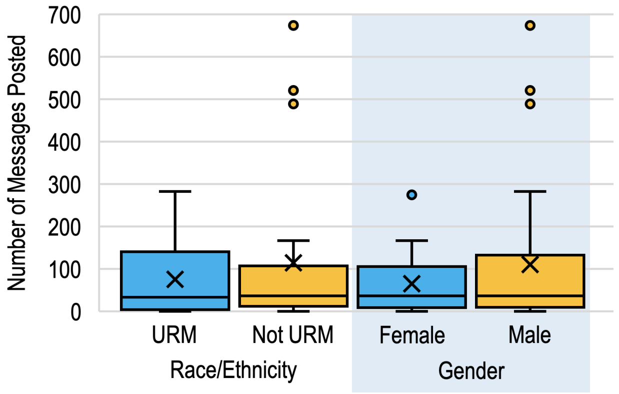

Figure 4: Number of messages posted by students who were from various underrepresented groups in computer science versus those who were not members of those groups. URM: M = 75.7, SD = 89.6; **Not URM:** M = 114.6, SD = 182.3. **Female:** M = 65.2, SD = 77.9; **Male:** M = 110.3, SD = 168.6.

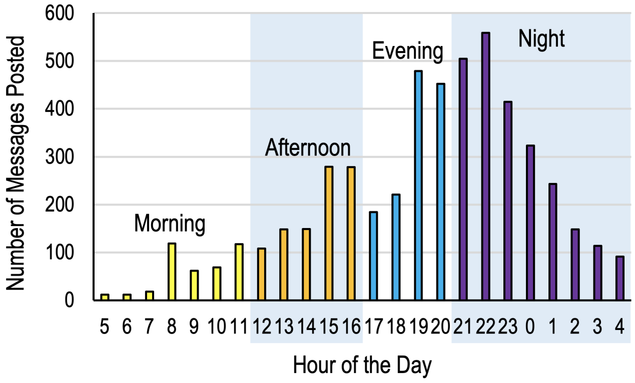

Figure 5: Number of messages students posted in Discord during each hour of the day.

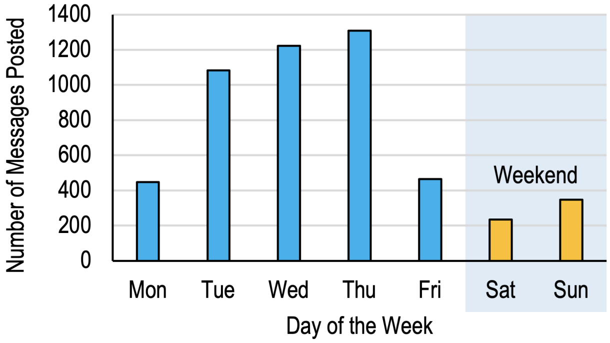

Figure 6: Number of messages students posted in Discord on each day of the week.

posted. In contrast, the weekend days, Saturday and Sunday, were the slowest, accounting for only 11% of all messages.

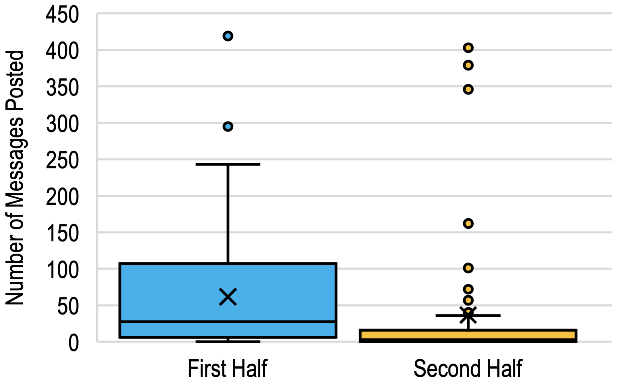

Figure 7: Number of messages students posted in Discord during the first half versus the second half of the semester. 1st Half: M = 61.4, SD = 83.5; **2nd Half:** M = 36.7, SD = 89.9.

4.6.3 Parts of the Semester. As Figure 7 shows, students posted noticeably more messages in Discord during the first half of the semester, when they were working on the skills assignments, versus the second half of the semester, when they were focused on the team software project. This difference was statistically significant (paired t-test: p = 0.04). Indeed, the number of messages students posted during the first half of course (3195) was 40% greater than the number they posted during the second half (1910)—a difference of 1285 messages. Moreover, roughly two-thirds of students (67%) posted fewer messages during the second half of the course, whereas less than a quarter (23%) posted more messages during the second half (with 10% posting the same number of messages during the first and second halves).

5 TAKEAWAYS AND DISCUSSION

5.1 24/7-Discord Approach Exceeded Goals

5.1.1 Increased Student–Teacher Interaction. Overall, the amount of help students received using our 24/7-Discord approach far exceeded that of prior semesters using Piazza with more limited teacher availability. The magnitude of the increase in student posts was particularly noteworthy (recall Figure 3)—for example, 90% of students posted a help message in Discord versus less than 50% during prior semesters using Piazza. Furthermore, two-thirds of the students posted 20 or more messages in Discord. Considering the prior work showing a positive correlation between student engagement in help forums and course success [2, 26], we found this level of engagement to be very encouraging.

5.1.2 Effective for Underrepresented Groups. An important concern when introducing a new educational intervention is the effect it will have on underrepresented groups, and our 24/7-Discord approach seems to have worked equally well for all types of students. We found no noticeable difference in the rates of posting between students from underrepresented groups and the other students (recall Figure 4). The only caveat to this observation is that, if females tend to post at a higher rate than males when they are able to post anonymously (as reported in [24]), then it is possible that our female usage was slightly depressed, making it more equal to the posting by males. Whether adding an anonymous posting option in Discord would increase posting by females is a question that would need to be addressed in future work.

5.1.3 Positive Feedback from Students. We were surprised and encouraged by the high volume of positive feedback we received from students in the course. We received student feedback from two main sources: a special Discord channel for providing and discussing feedback (not anonymous) and the student evaluation of teaching survey administered by the university (anonymous). Although our 24/7-Discord approach to help delivery was not the only aspect of the course that students liked, they mentioned the approach in many of their comments. In particular, several students specifically mentioned liking Discord, for example:

> "discord was extremely helpful this semester"

> "i will say that. Discord is a far better platform than zoom or any other alternative. Even when you go back to campus you should still create a discord server for your courses."

However, even more students commented favorably about the availability of round-the-clock help, for example:

> "katie and jeff are the best TAs ive ever had in a class. Maybe its because of discord with them being so easily accessible, but jeez they're great."

> "I couldn't agree more about the TAs. I can literally reach them anytime of the day and any day of the week."

> "thankfully the TAs were ready basically 24/7 to help me when I needed the assistance"

Furthermore, students responded very positively to this Likert-style question from the university's teaching evaluation survey: "I had an opportunity to ask questions in or outside of class" (M = 4.89 out of 5, N = 36).

5.2 When Students Sought Help

5.2.1 Late Evening/Nighttime Was Peak Time. By far, the peak time for students to post help messages was late evening and at night (recall Figure 5). For example, 65% of messages were posted between 7:00 p.m. and 5:00 a.m, with 47% coming between 7:00 p.m. and midnight. Interestingly, our results somewhat contradict those from a prior study [33] in which only 1% of student activity occurred between midnight and 5:00 a.m.—in contrast, 18% of student posts in our course happened during those hours. It remains an open question as to why this discrepancy occurred.

5.2.2 Spikes Right before Deadlines. As described in Section 2.1, numerous prior studies have found a tendency among CS students to procrastinate on their work until right before deadlines, and our students appear to have been no different. In particular, the spikes in message posting activity on Tuesdays, Wednesdays, and Thursdays (recall Figure 6) likely occurred because those were the days on which most deadlines fell. Anecdotally, the teachers can confirm this assertion, recalling noticeable increases in demand for help just before deadlines.

5.2.3 Less Activity on Weekends. We were somewhat surprised by the reduced message-posting activity on weekends (recall Figure 6). We had rather assumed that undergraduate students busy with multiple courses would tend to use weekends to catch up on coding-intensive assignments that tend to require considerable effort and concentration; however, this did not appear to be the case for most students, with only 11% of messages being posted on the weekends.

5.2.4 Fewer Posts during the Project. There was a noticeable decrease in student help posts between the first half of the course, which focused on weekly skills assignments, and the second half, which focused on collaborative development work (recall Figure 7), and we have a few ideas as to why that might have been. Anecdotally, the teachers observed that when problems were more well structured, as they were during the first half of the course, students were more able to explain their problems and to understand the answers that the teachers provided using only text messaging. However, for problems that were more open ended, as they were during the project portion of the course, students preferred to use voice calling and screen sharing to explain their questions and to receive the teachers' answers. Because we were unable to collect data on voice-channel usage, our results may have missed this increase in help being delivered on those channels. There are also a number of potential reasons why students' need for teacher help may have simply decreased during the second (team-project) half of the course: general learning effects, receiving help from teammates instead of teachers, and fewer deadlines during the project, so procrastinating students may have had fewer windows of time where they were actively working.

6 CONCLUSION

In this experience report, we described the use of an approach to delivering round-the-clock help to students in a remote software engineering course using Discord. The approach exceeded our expectations for increasing the amount of help students in the course receive from teachers. Moreover, students from underrepresented groups interacted with teachers as much as other students, and we received consistently positive feedback from students regarding the approach. In applying the approach, we also learned that students tended to seek teacher help the most during late evening and nighttime, right before deadlines, and during the first half of the course.

Looking to the future, more research would be needed to address some limitations of our 24/7-Discord approach and of our findings about it. For instance, a key challenge in applying our approach is its demand on human resources (i.e., teachers)—in particular, it requires many hours of teacher time, far exceeding the 20 hours per week that a single TA typically works. One possible solution would be to focus on making help available during the peak times—for example, between 7:00 p.m. and midnight (or later) based on our findings. Another possible solution would be to pool TAs by sharing them across courses, similar to Campbell and Craig's drop-in help centre approach [3]. Future studies would also be needed to disentangle the relative impact of Discord versus the 24/7 availability of teacher help, to better understand students' usage of voice/video features for receiving help (for which we were not able to collect data), and to investigate the extent to which our findings generalize (after all, this was an experience report regarding a single course). In conclusion, our strong results provide compelling motivation for future research on our 24/7-Discord approach and other such approaches for improving the quality and quantity of teacher help that students in technically challenging CS courses receive.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1822816 and Grant No. 1918751. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Abeer AlJarrah, Michael K. Thomas, and Mohamed Shehab. 2018. Investigating temporal access in a flipped classroom: procrastination persists. *Int. J. Educ. Technol. High. Educ.* 15, 1 (2018). https://doi.org/10.1186/s41239-017-0083-9

[2] Debarati Basu, Sarah Heckman, and Mary Lou Maher. 2021. Online Vs Face-to-Face Web-Development Course: Course Strategies, Learning, and Engagement. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. 1191–1197. https://doi.org/10.1145/3408877.3432438

[3] Jennifer Campbell and Michelle Craig. 2018. Drop-In Help Centres: An Alternative to Office Hours. In *Proceedings of the 23rd Western Canadian Conference on Computing Education (WCCCE '18)*. Article 9. https://doi.org/10.1145/3209635.3209642

[4] Campuswire 2021. *Campuswire*. Retrieved December 2, 2021 from https://campuswire.com/

[5] Allan Collins, John Seely Brown, and Ann Holum. 1991. Cognitive Apprenticeship: Making Thinking Visible. *American Educator* 15, 3 (1991), 6–11.

[6] Tom Drange and Joakim Kargaard. 2017. Increasing student/student and student/lecturer communication through available tools to create a virtual classroom feeling in online education. In *Proceedings of the 13th International Scientific Conference eLearning and Software for Education*. 393–400. https://doi.org/10.12753/2066-026X-17-058

[7] Armando Fox and David Patterson. 2012. Crossing the Software Education Chasm. *Commun. ACM* 55, 5 (May 2012), 44–49. https://doi.org/10.1145/2160718.2160732

[8] Ria Galanos, Whitaker Brand, Sumukh Sridhara, Mike Zamansky, and Evelyn Zayas. 2017. Technology We Can't Live Without! Revisited. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 659–660. https://doi.org/10.1145/3017680.3017691

[9] Daniel D. Garcia, Eric Allatta, Manuel Pérez-Quiñones, and Jeff Solin. 2015. Technology We Can't Live Without!. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. 597–598. https://doi.org/10.1145/2676723.2677336

[10] Philip Johnson. 2019. Design and Evaluation of an "Athletic" Approach to Software Engineering Education. *ACM Trans. Comput. Educ.* 19, 4, Article 41 (Aug. 2019). https://doi.org/10.1145/3344273

[11] Philip M. Johnson, Hongbing Kou, Joy M. Agustin, Qin Zhang, Aaron Kagawa, and Takuya Yamashita. 2004. Practical Automated Process and Product Metric Collection and Analysis in a Classroom Setting: Lessons Learned from Hackystat-UH. In *Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE '04)*. 136–144. https://doi.org/10.1109/ISESE.2004.1334901

[12] Ayaan M. Kazerouni, Stephen H. Edwards, and Clifford A. Shaffer. 2017. Quantifying Incremental Development Practices and Their Relationship to Procrastination *(ICER '17)*. 191–199. https://doi.org/10.1145/3105726.3106180

[13] Lisa Lacher and Cydnee Biehl. 2018. Using Discord to Understand and Moderate Collaboration and Teamwork: (Abstract Only). In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. 1107. https://doi.org/10.1145/3159450.3162231

[14] Lisa Lacher and Cydnee Biehl. 2019. Investigating Team Effectiveness Using Discord: A Case Study Using a Gaming Collaboration Tool for the CS Classroom. In *Proceedings of the 15th Int'l Conf on Frontiers in Education: Computer Science and Computer Engineering (FECS '19)*. 199–204.

[15] David J. Malan. 2009. Virtualizing Office Hours in CS 50. In *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '09)*. 303–307. https://doi.org/10.1145/1562877.1562969

[16] Kenrick Mock. 2019. Experiences Using Discord as Platform for Online Tutoring and Building a CS Community. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. 1284. https://doi.org/10.1145/3287324.3293769

[17] National Science Foundation, National Center for Science and Engineering Statistics. 2017. *Women, Minorities, and Persons with Disabilities in Science and Engineering: 2017.* Special Report NSF 17-310. Arlington, VA. Available at https://www.nsf.gov/statistics/2017/nsf17310/.

[18] Claudia Ott, Brendan McCane, and Nick Meek. 2021. Mastery Learning in CS1 - An Invitation to Procrastinate?: Reflecting on Six Years of Mastery Learning. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE '21)*. 18–24. https://doi.org/10.1145/3430665.3456321

[19] Thomas H. Park and Susan Wiedenbeck. 2011. Learning Web Development: Challenges at an Earlier Stage of Computing Education. In *Proceedings of the Seventh International Workshop on Computing Education Research (ICER '11)*. 125–132. https://doi.org/10.1145/2016911.2016937

[20] Thomas W. Price, Zhongxiu Liu, Veronica Cateté, and Tiffany Barnes. 2017. Factors Influencing Students' Help-Seeking Behavior While Programming with Human and Computer Tutors. In *Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17)*. 127–135. https://doi.org/10.1145/3105726.3106179

[21] Yanyan Ren, Shriram Krishnamurthi, and Kathi Fisler. 2019. What Help Do Students Seek in TA Office Hours?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 41–49. https://doi.org/10.1145/3291279.3339418

[22] Clifford A. Shaffer and Ayaan M. Kazerouni. 2021. The Impact of Programming Project Milestones on Procrastination, Project Outcomes, and Course Outcomes: A Quasi-Experimental Study in a Third-Year Data Structures Course. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. 907–913. https://doi.org/10.1145/3408877.3432356

[23] Jaemarie Solyst, Trisha Thakur, Madhurima Dutta, Yuya Asano, Andrew Petersen, and Joseph Jay Williams. 2021. Procrastination and Gaming in an Online Homework System of an Inverted CS1. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. 789–795. https://doi.org/10.1145/3408877.3432440

[24] Adrian Thinnyun, Ryan Lenfant, Raymond Pettit, and John R. Hott. 2021. Gender and Engagement in CS Courses on Piazza. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)*. 438–444. https://doi.org/10.1145/3408877.3432395

[25] Arie Van Deursen, Maurício Aniche, Joop Aué, Rogier Slag, Michael De Jong, Alex Nederlof, and Eric Bouwers. 2017. A Collaborative Approach to Teaching Software Architecture. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 591–596. https://doi.org/10.1145/3017680.3017737

[26] Mickey Vellukunnel, Philip Buffum, Kristy Elizabeth Boyer, Jeffrey Forbes, Sarah Heckman, and Ketan Mayer-Patel. 2017. Deconstructing the Discussion Forum: Student Questions and Computer Science Learning. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 603–608. https://doi.org/10.1145/3017680.3017745

[27] Monica Vladoiu and Zoran Constantinescu. 2020. Learning During COVID-19 Pandemic: Online Education Community, Based on Discord. In *Proceedings of the 19th RoEduNet Conference: Networking in Education and Research (RoEduNet '20)*. 1–6. https://doi.org/10.1109/RoEduNet51892.2020.9324863

[28] Jennifer Wang and Sepehr Hejazi Moghadam. 2017. Diversity Barriers in K-12 Computer Science Education: Structural and Social. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. 615–620. https://doi.org/10.1145/3017680.3017734

[29] Wikipedia 2021. *Discord (software)*. Retrieved July 22, 2021 from https://en.wikipedia.org/wiki/Discord_(software)

[30] Wikipedia 2021. *Elluminate Live*. Retrieved August 11, 2021 from https://en.wikipedia.org/wiki/Elluminate_Live

[31] Wikipedia 2021. *Piazza (web service)*. Retrieved August 11, 2021 from https://en.wikipedia.org/wiki/Piazza_(web_service)

[32] Wikipedia 2021. *Slack (software)*. Retrieved August 10, 2021 from https://en.wikipedia.org/wiki/Slack_(software)

[33] Albina Zavgorodniaia, Raj Shrestha, Juho Leinonen, Arto Hellas, and John Edwards. 2021. Morning or Evening? An Examination of Circadian Rhythms of CS1 Students. In *Proceedings of the IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET '21)*. 261–272. https://doi.org/10.1109/ICSE-SEET52601.2021.00036