CHAPTER 8

A Model for Mentoring Faculty and Teaching Assistants in Active Learning

CELINE LATULIPE AND STEPHEN MACNEIL

Moving to an active learning mode of teaching requires a fair amount of change across a variety of dimensions. There is a mental mindset adjustment, changes to class structure, prep work to develop, new tools to learn, and new ways of interacting with teaching assistants (TAs), who can range from sophomore undergraduates to graduate students. Because this type of teaching can be so different, we propose that a collaborative/interactive model at multiple levels is the best way to bring faculty and TAs on board for a particular course. For inexperienced instructors, a team-teaching model is employed so that more senior teaching faculty can help scaffold the teaching experience for less experienced faculty members. By sharing the workload, new classes can be created quickly to address emerging needs within the department. At the same time, pedagogies, best practices, and important tacit knowledge can be shared among instructors, a way to apply active learning to the act of education itself.

TAs face the same challenges as they transition into teaching roles. Departments and university systems often do not provide explicit training for new TAs. There is an assumption that new TAs will shift effortlessly from their current role as students to being able to teach, grade, and interact with students effectively. This puts a lot of pressure on new TAs and can reduce their potential effectiveness. Often, through informal mentorship, TAs learn how to perform their roles and then improve over time as they gain experience. By designing the graduate and undergraduate teaching team with formal mentorship, it is possible to ensure a smoother transition for new TAs, reduce the load on the instructor to personally mentor each new student, and provide more senior TAs with valuable mentorship experience.

Both formal and informal mentorship offer an important scaffold to help new teaching team members develop their skills and transition into their new roles. By adopting these two models of mentorship, the stepping stone model (Roberts, Kassianidou, & Irani, 2002) and team teaching, we are able to create a better teaching environment. These models have helped us to reduce the workload, provide opportunities for learning and skill development, and ensure that tacit knowledge is preserved across multiple semesters for all members of the teaching team.

The work described in this chapter represents practices developed in fully flipped active learning (FFAL) courses in the College of Computing and Informatics (CCI) at UNC Charlotte. FFAL is a teaching approach in which students learn the material out of class and then practice and apply the material during class time. This practice is contrasted by partially flipped active learning, where lecture and activities are intertwined, and students learn some material out of class. Examples of the use of the mentoring model are largely drawn from the Data Structures and Algorithms course discussed in a previous chapter as well as in prior published research (Latulipe, MacNeil, & Thompson, 2018). In that class, we evolved a self-sustaining TA mentoring cycle over time that has allowed for a rich and personal learning experience for students, even in large classrooms. All TAs, from sophomore undergraduates to graduate students, learn from each other and stay with the class sometimes up to five semesters. The continuous cycle of new undergrad TAs coming in and learning from the more senior TAs helps with continuity. The faculty member, Dr. Latulipe, strives to empower all of the TAs with responsibility over various aspects of the course and welcomes ideas for course improvement from all involved. The TAs get a much richer experience than if they were just grading assignments, and they experience a cycle of being mentored by more senior TAs and then mentoring the junior TAs coming in.

A Case for Mentorship

Developing a professional identity is challenging across majors and disciplines. Students' reasons for choosing to major in computer science (CS) and for staying in the program are complex; however, there is some evidence that their interactions with other students strongly affect these decisions (Bean, 2005). To help students feel more comfortable in their major, it is important for them develop their self-efficacy (Barker, McDowell, & Kalahar, 2009) and their identity as a computer scientist (Lewis, Yasuhara, & Anderson, 2011). Students with self-efficacy believe that they can accomplish their goals, and self-efficacy is related to the concept of growth mindset (Dweck, 2008) and students' ability to bounce back from failures. This resilience is an important predictor of academic persistence and the ability to do well in difficult programs, especially important in science, technology, engineering, and mathematics (STEM) fields like computing, where some students are significantly more prepared and have more prior experience than their peers (Latulipe et al., 2018). The other important aspect to consider is students' identity as a computing professional. Identities are socially constructed based on our interactions with others. Stereotype threat and intended or unintended microaggressions can serve to undermine this identity. Students may look around the classroom or at big technology companies and see a lack of diversity and internalize this as a message that they are not welcome in the field. Similarly, students may assume behaviors consistent with their gender to fit in with their peers. Finally, microaggressions are not uncommon in science classrooms. Many classes promote competition over collaboration based on the ways that students are graded or on the way interactions between students are designed (accidentally or otherwise). These classrooms can lead to "negative stereotypes about CS, such as it being competitive, singularly-focused, asocial, and primarily male" (Lewis, Anderson, & Yasuhara, 2016).

These challenges can be addressed through mentorship in the classroom. We propose using a combination of faculty and student mentors from our teaching team. If students have multiple sources of mentorship in the classroom, they are more likely to see someone like them on the teaching team (similar race, age, gender, interests, etc.). At the same time, students are more likely to see a progression of skills between themselves and the teaching faculty. Teaching faculty often teach the same class repeatedly; they have obtained a PhD, and they have decades or more of research experience in the field. Students do not see this progression when interacting with faculty members. Students may become discouraged and feel that they too will need to obtain a similar amount of experience to be successful. This can be overwhelming for students who are new to the field, but this level of experience is not required for all computing professionals. By providing multiple types of mentors (faculty, graduate TAs, and undergraduate TAs), students are more likely to see how developing skills in computing is a long and persistent process. Students can attribute setbacks to that aspect rather than to their perceived intellectual limitations. Finally, multiple levels of mentorship provide students with a balance of technical, emotional, and interpersonal support at multiple levels in the teaching team and even encourage students in the class to engage in informal mentorship with each other. This model is called the stepping stone model of mentorship.

Stepping Stone Model of Mentorship

The stepping stone model of mentorship was introduced by Roberts et al. (2002) who advocate for having different mentors for students at different levels within the university. For example, students can be mentored by faculty or by other students more senior in the department. They suggest that doing so helps students to look "not 20 years to the point at which they might themselves be a professor, but one year to the point at which they might become a section leader" (Roberts et al., 2002). While all students may not intend to become professors, they can see that expertise in computing is a continuum and that through persistent practice they can develop these skills over time. The stepping stone model helps students make short-term, achievable goals. The authors cite this model as being critically important for increasing the participation of women in computing. While we adopt this stepping stone model of mentorship, we also consider how mentoring can happen among the mentors themselves and not just between the mentors and the students. Finally, unlike the original stepping stone mentorship model, we implement this model in individual courses rather than throughout the curriculum and department. In Figure 8.1, we present our model for mentorship that is heavily influenced by the stepping stone model of mentorship. By adding in this aspect of intermentor mentorship, we find that our course runs more smoothly. TAs are exponentially engaged and proud of the work that they do, and they learn many more skills than they would otherwise. In this chapter, we describe this mentorship model and how we implemented it in one of our classes. We close with a brief evaluation to demonstrate its effectiveness as a classroom organization technique.

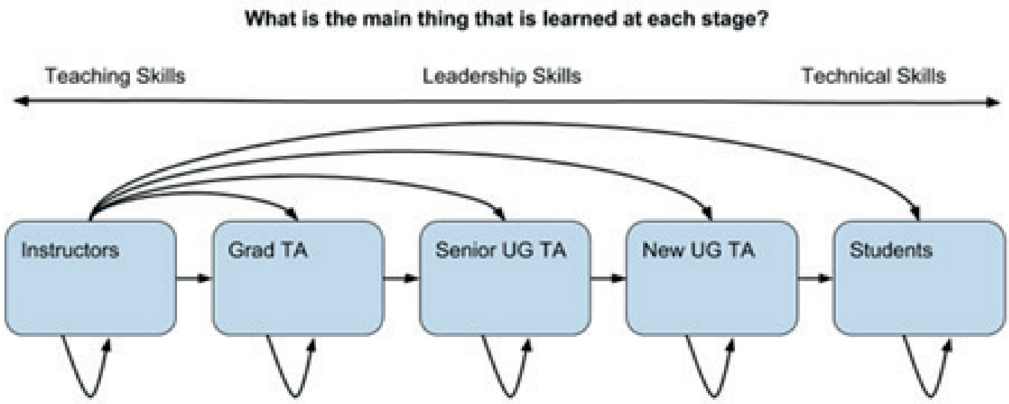

Figure 8.1. A modified stepping stone model of mentoring. The arrows that loop back to the same cells represent the idea that students may mentor each other, and, if you have multiple senior undergraduate (UG) teaching assistants (TAs), they may also mentor and teach one another.

An important aspect of the stepping stone mentorship model is the idea that each individual is mentored by someone who is close to them in terms of the next step they would take in their personal or career progression. So, having a sophomore student mentored by a TA who is a junior or senior undergraduate helps the student not only learn the material but it also allows them to see how students behave and what they know how to do at the next level. A senior TA probably does not need to learn about the higher-level course management from the professor; that is likely more salient to the doctoral-level TA, who is potentially going to move to some other institution and be in charge of their own courses.

Faculty-to-Faculty Mentoring

Mentoring faculty about active learning teaching practices has been a major thrust of effort in the CCI. In this section we highlight two aspects of this: pair teaching and the role of the Center for Education Innovation in creating a supportive community of practice.

Pair Teaching

One of the practices that we have found very beneficial in our college is the use of pair teaching for newly flipped classes. We have three cases of classes that were previously taught in a traditional lecture format but have been completely restructured as FFAL classes. One of these classes is described in detail in Chapter 3: A Fully-Flipped Active Learning Course. The amount of work required to make this major transition is tremendous; typically, a summer is devoted to preparation of materials, creating a structure, and setting up the technology needed for the class. In all three cases, we had two faculty involved the first semester. Involving two faculty in the course transformation process helps split the workload. The two faculty involved also benefit from talking through the design process with each other, generating more ideas and solutions than one faculty member working alone.

A major benefit of this pair teaching approach allows a faculty member with expertise in flipping classes and teaching classes in the flipped style to share that expertise with another faculty member. While this may be seen as expensive in terms of teaching resources, the cost is only for one semester, and it has the effect of increasing the active learning teaching capacity within a department.

In our CCI, an FFAL version was created for the very first programming course, and two instructors (Associate Professor Latulipe and Senior Lecturer Long) were involved in that development and in teaching the initial offering. Both Latulipe and Long then worked to develop a fully flipped model of the subsequent programming class. Long continued to teach the second class as an FFAL class and then cotaught it with an associate professor to help her learn the FFAL approach. The associate professor has since gone on to create an FFAL version of the third programming course at the sophomore level. When developing and teaching it for the first semester, they worked with Associate Dean Perez-Quinones, who had no direct experience with the FFAL approach, but had domain knowledge having taught a similar programming course for many years. Having an associate dean involved also helped the college administration gain a better understanding of the challenges and benefits of active learning. Long continues to teach that class as an FFAL course.

Community of Practice

Our college has also developed a series of Summer Institute for Active Learning programs to help our faculty share how to teach computing courses this way. These summer institutes are run through the Center for Education Innovation within the CCI. This center and the summer workshops and pedagogy luncheons throughout the year have served to create a supportive community of practice in which faculty can mentor each other (Frevert et al., 2018; Maher et al., 2016). Indeed, we have applied the stepping stone model of mentorship to the way the Summer Institutes are run and have assigned faculty members who have previously participated in the workshops to be senior mentors.

Faculty-to-faculty mentoring within our college has helped our faculty learn about the process of flipping a course, about the tools used for supplying appropriate prep work to students and for conducting active learning in the classroom, and about how to structure teams and course work to scaffold student success. However, the mentoring also allows the more experienced FFAL faculty to share their philosophy about teaching and their attitude toward students. The FFAL faculty in our college have engaged in a practice of data-driven responsive teaching. Instead of throwing up our hands in frustration and saying things like: "These students coming in don't know anything! How can I be expected to teach them Z when they don't know X and they can't do Y," our faculty look at the backgrounds of students coming into the classes and then share that aggregate data with the class. They express an attitude that says: "You may not have had the same exposure to this material as someone sitting beside you, but that doesn't mean you are less capable of becoming a good software developer." We believe that this attitude and the exploration of the data around student backgrounds and preparations help our faculty.

All of this faculty mentorship has led to much more collaborative teaching in some of our higher-level core classes. For example, we now have three faculty members teaching three sections of our Networking and Operating Systems class, and all of them are teaching the class as FFAL, sharing resources and even having joint discussion forums for students across the sections. We also have two faculty members teaching sections of Software Engineering and sharing active learning resources and content. Similar arrangements are happening across some sections of our introductory programming courses. This collaborative teaching enhances the student experience because the faculty are able to do more active learning when the activities are jointly created and shared.

TA Mentorship Cycle

While mentorship among faculty can help spread active learning practices, the role of TAs in supporting active learning in classrooms cannot be overlooked. TAs can make a major difference in the student learning experience. In this section, we describe how we have created a cycle of mentoring across various levels of TA (graduate, senior undergraduate, junior undergraduate) experiences.

One of the issues with teaching at the college level is that faculty members might get assigned a new TA every semester and not think that they have any say in the matter. Such a high turnover rate leads faculty members to not consider how best to make use of TAs in their classes and so regard the TAs as simply graders. This means that the work experience for the TA is not engaging and, in fact, is mostly seen as unimportant drudgery. Thus, many TAs do not stay with a course longer than one semester and search for newer and better opportunities. By providing more enriching teaching experiences that allow them room for professional development, we can retain these TAs and ensure better continuity in the course from semester to semester. This continuity is important because a lot of tacit knowledge is hard to document (or is very time consuming to document), especially in courses that are flipped and have extensive active learning components.

TA Team Responsibilities

In the Data Structures class, the TA team typically consists of three to seven undergraduate hourly TAs, depending on the size of the class. Under the guidance of the professor, the TAs are responsible for different aspects of the course. These responsibilities may include testing in-class lab activities before class, reviewing test questions, checking students off as they complete labs, developing or testing the individual programming assignments, grading assignments, setting up assignments in the autograding system, running submitted programming assignments through the plagiarism detection system, answering questions on the class discussion forum, planning and conducting help sessions (especially around the time that assignments are due), and proctoring tests. A TA typically runs the podium laptop with the

Poll Everywhere quiz while the instructor moves around the classroom. The TAs have weekly meetings with the professor to keep the class on track, divide up grading, brainstorm class improvements, and deal with any issues that come up. Because each TA takes charge of a few of these aspects of the class, they feel some ownership in the success of the class as a whole, and that adds to the integrity of the classroom experience.

TAs in the two most recent semesters have developed their own infrastructure. They have set up a code version management repository for managing the testing and refinement of assignments. They have also set up Discord servers with different channels for faculty and TAs to conduct asynchronous discussion of assignments, labs, tests, etc. The Discord server helps manage tasks and facilitate communication, but it also serves as a platform for discussion of new ideas to improve the class. It has become a small online community, and the TAs and faculty end up supporting each other through this medium. At the end of one semester, one of the TAs announced to all on the Discord server that they were having a games day at their place and the other TAs were all invited. This demonstrates that the TA team can become a cohesive community.

Many classes in CCI use a discussion forum to encourage conversations that extend beyond the classroom. We use Piazza as our discussion forum software because it integrates with our learning management system that students use to submit assignments and obtain course material. While the professor and all the TAs generally monitor the Piazza class discussion forum, one of the TAs is typically assigned to decide a schedule for when a TA is going to monitor the Piazza discussion forum for each weekend over the semester so that everyone takes turns. The person in charge of monitoring Piazza on any given weekend is not in charge of answering every question but rather alerting another TA or the faculty member if an issue comes up that they need to address.

One of the most critical roles of TAs in the Data Structures class is running weekly help sessions. These sessions may give more in-depth examples of some of the programming skills needed or how to make use of some of the digital tools that are part of the course. During weeks when assignments are due, the TAs help students understand the assignment requirements, help talk students through approaches to finding solutions, and sometimes help students debug their code. Thus the review sessions not only serve as a source of technical help, but also they often turn into collaborative work sessions for students. Students in the help sessions have been observed to figure out a problem and then go and help other students who are having the same problem. This is an indication of the comfort in the classroom, and it is also an indication that our model for mentorship is adopted informally among students in the class. Students can get to know other students in these help sessions, so they become a source of friendship, in part because the TAs work hard to create a casual, safe space.

TA Mentoring: Instructor to Graduate

Instructors help new graduate TAs move into the position of teaching the class by showing them how to manage the class, generate new material, and guide the teaching staff. They also spend a significant amount of time discussing teaching philosophy, especially why they are teaching the class as an FFAL class. The professor and graduate TAs have informal discussions about the various challenges, how data are collected to make evidence-based improvements, and how the teaching experience differs from a traditional lecture class. These discussions occur because graduate TAs are often involved in codesigning the material and the formative and summative assessments. Fostering the graduate TAs' understanding of these classroom design decisions has a ripple effect: they are able to advocate for these aspects to students and other TAs. This is also helpful for the graduate student TA, especially if the student is a doctoral student who is planning a career in academia. It means the doctoral student can write a much more detailed and sophisticated teaching philosophy statement for their job search.

TA Mentoring: Graduate to Undergraduate

Graduate TAs (GTAs) typically have more experience with course content and more technical skills than undergraduate TAs, and that can be useful. However, finding or training good GTAs can be challenging. They do not typically have a lot of time to invest in their TA work, and they may feel pressure from their research advisor to spend as little time as possible on TA duties to maximize the time they spend on their research. These signals may also inadvertently or purposefully tell doctoral students that teaching is not important. Also, most doctoral students would have taken a core course taught in a traditional lecture style. They may not buy into the active learning approach and may even suffer from typical academic nostalgia: "I had to suffer through boring lectures and hard assignments with no help, so these students should, too." Such GTAs may not be thinking about inclusive education and supporting diverse student success and therefore may not have the appropriate attitude needed for the commitment involved in teaching an FFAL class.

On the other hand, some GTAs are more inclined to consider teaching important, and these doctoral students are often enrolled in the university teaching certificate. They have a more positive attitude about their classroom-based responsibilities and really cherish the opportunities they are given to share ideas and to work directly with students. Finding great GTAs, mentoring them, and providing them with opportunities for growth as teachers can be rewarding, especially as they mentor undergraduate TAs.

There may also be GTAs who fit between these two ends of the spectrum: graduate students who think teaching is important and are interested in pedagogical innovation but are overwhelmed by their other research work and responsibilities. These graduate students could also play a role but may need more mentoring, especially with respect to time management. The ability of these GTAs to be assisted in the workload by the use of undergraduate TAs may help them see the power in the stepping stone teaching mentorship model.

Good GTAs can pass on knowledge about the content and can teach undergraduate TAs how to use the grading systems, the learning management system (including the functionality that comes with the TA role), and various other technologies that might be used in the course. In the Data Structures course, one particularly good GTA helped the undergraduate TAs learn how to set up programming assignments in the Web-Cat autograding system and how to create and share things like lab checkoff Google forms. GTAs can also lead help sessions with senior undergraduate TAs, teaching them the best way to help students. In this case, the undergraduates may actually help the GTAs to ensure they are explaining things at the right level. Sometimes the GTAs are so far removed from having been an undergraduate that they use too much jargon and give too much detail. The undergraduate TAs can check them on that and ensure that the help sessions are at the right level for the undergraduate students.

TA Mentoring: Undergraduate Senior to Undergraduate Junior

Although GTAs are often more knowledgeable and better organized than undergraduate TAs, the undergraduate TAs often have much more time and energy. For instance, undergraduate TAs may be more inclined to stay after class and provide help sessions to socialize with students, helping students feel more comfortable and seeing the teaching staff as personable and approachable. This strength can also be a double-edged sword as they are constantly hungry for new and exciting opportunities. We have designed a progression that continually challenges undergraduate TAs with new and exciting tasks and roles, starting off with well-scoped and highly-structured work to ensure that they are successful and to avoid overwhelming them, each semester assigning more responsibilities and changing the types of roles that they perform.

One of the main roles that junior undergraduate TAs (JUTAs) perform is to debug the labs and test cases before the students in the class attempt them. New TAs are often excited about this kind of work, which would be less exciting for more senior undergraduate TAs and graduate TAs. It also gives them an opportunity to practice the material and prepare for the labs. The JUTAs often challenge themselves to create more elegant solutions than they would have submitted as students. It can be a rewarding experience for these JUTAs to breeze through assignments and labs that used to take them hours and even days. JUTAs are often likely to suggest novel ideas about how to change the class because they have most recently experienced it as a student, and they are less inhibited by "what is possible" in a class. More senior TAs may realize how difficult an idea is to implement before exploring it further. New TAs often do not know how hard it is to make changes, and so they suggest more radical ideas, which can often lead to interesting changes when adapted by more senior TAs and faculty who have a more pragmatic perspective. Finally, the JUTAs are given one important area of responsibility that is overseen by a more senior TA. Having ownership over some aspect of the course gives JUTAs pride about their work, and we have even observed cases of JUTAs bragging to students in the class about something that they implemented, managed, or fixed in the class structure or class material. This is a really good recruitment tool to get new students interested in joining the teaching staff.

The next aspect of our mentorship model involves the senior undergraduate TAs (SUTAs). To maintain undergraduate TAs' enthusiasm and excitement, we have carefully designed a progression for JUTAs to slowly take on new challenges as they become more senior in the teaching team. After a semester or two, the enthusiasm and excitement of being an undergraduate student begins to become a driving force for them to want to try something new. At this point, many undergraduate students leave their TA role to search for new and exciting challenges. While understandable, this can be disruptive for a teaching team. Our mentorship model was designed with this in mind. In our mentorship model, we attempt to provide different experiences for SUTAs. This maintains their enthusiasm and excitement as they begin to develop new skills and have new challenges. The main focus for SUTAs is to train and pass on their knowledge to JUTAs. They also provide supervision to help ensure that JUTAs are successfully developing skills and performing their duties. These management skills are important for when SUTAs start applying for jobs because it is a new leadership challenge for many students. Often the only leadership skills that students have at this point in their careers include group work in class and maybe some leadership roles for student organizations, so being an SUTA is a unique and exciting role for them. SUTAs are typically supervised by the graduate TAs, but they also interact frequently with the faculty members of the teaching team.

We have described how our scaffolding the experience of undergraduate TAs is not only vital for the success of teaching teams but also a mentorship model that focuses on providing balance and progression for students. JUTAs have unique benefits that they provide to the teaching team, and the roles and tasks that they perform help to ensure that they are learning new skills and feeling like an important part of a team. They are afforded opportunities to exercise agency in their role and have their own projects; however, the projects are supervised and structured in a way to ensure success. SUTAs remain engaged in the team, they develop leadership skills, and they start to train their JUTA replacements. Training their replacements ensures continuity in our mentorship process.

Student Mentoring: Teaching Team to Students in Class

The final aspect of our mentorship model is the relationship between the teaching team and the students in the class. Each of the different members of the teaching team has a different relationship with the students that is carefully designed to help the students and leverage the unique aspects of the teaching team. We have already described previously the different aspects of the teaching team; in this section we describe their relationships with the students.

The JUTAs are closest in experience and age to the students in the class. In our mentoring cycle, they often ask to become TAs because they have been in the class and have been helped by JUTAs. JUTAs provide a role model for becoming a TA in the class and the importance of working hard and learning the material. They can provide relatable stories about their own struggles to learn the material and how they overcame setbacks. They can also relate to student misconceptions and misunderstandings with the material. In this way, they are very approachable for students in the class. JUTAs often also stick around after class and after help sessions to socialize and discuss topics outside of the course material. In this way, they can serve as friends to the students in the class. The JUTAs are a very important part of our stepping stone mentorship model. They embody the first step in a progression from student to faculty. Their impartial and imperfect understanding of concepts can actually be a benefit in convincing students to adopt a growth mindset. Students may look at an instructor and think that they have always been experts in the field of computing, but JUTAs give students a relatable example that expertise is developed through hard work and practice.

SUTAs are more removed from the students' experiences; however, they may still serve many of the same roles that JUTAs serve. Because SUTAs are more removed, they can also be more impartial when interacting with students. This can be very beneficial to ensure that all students are being treated fairly. SUTAs are often managing important parts of the course, such as the autograding system and writing test cases and assignments. Thus, they are more familiar with the specifics of the assignments, an impartiality that can be beneficial; at the same time, SUTAs are still very familiar with the challenges that students face. They may be more willing to advocate for students when an assignment is difficult or a deadline should be extended. Finally, as SUTAs begin to search for jobs and other opportunities, they can provide students with important information about what additional skills should be developed, how to frame the work they do in class to impress recruiters, and about different possible careers. Often SUTAs are asked by students to stay after class to give advice about applying for internships and jobs. This is a testament to the respect that students have for SUTAs.

The GTAs are the most experienced and often the most technically sound members of the teaching team. They have completed their undergraduate degrees and are actively doing research. This allows them to solve technical problems in the class that might be too difficult or time consuming for the rest of the teaching team. They can also help students in cases where the SUTAs and JUTAs are unable to figure out a solution on their own. GTAs can also be too removed from the undergraduate students' experiences. They can be overly technical, and they can use a lot of unfamiliar jargon. For some advanced students in the class who have significant previous programming experience, this can be exciting. It is not uncommon for more experienced students in the class to gravitate to the GTAs and to ask them for advice about hobbyist projects, graduate school, and careers in research. These students, who might otherwise be bored by the assignments, can still be motivated to attend class and help sessions to interact with the GTAs.

The instructor oversees and is responsible for all aspects of the course but delegates some of the operational aspects to TAs at various levels. The instructor is able to help students with technical problems but can often also design videos and exercises that present the material in a way that is fun and engaging. They can make the classroom environment enjoyable for students, but they can also be strict and impartial when needed to ensure that all students are treated fairly. The instructor is responsible for communicating to students important aspects at play in the classroom. They can relate the material to real-world examples, but they can also discuss important issues in technology, such as sexism, racism, and bullying and how privilege plays out in the educational system. The instructor can also present concepts from learning sciences research, such as the growth mindset and constructivist theories of cognitive development, to ensure that students have a better understanding of why their learning environment has been designed as a flipped classroom with active learning.

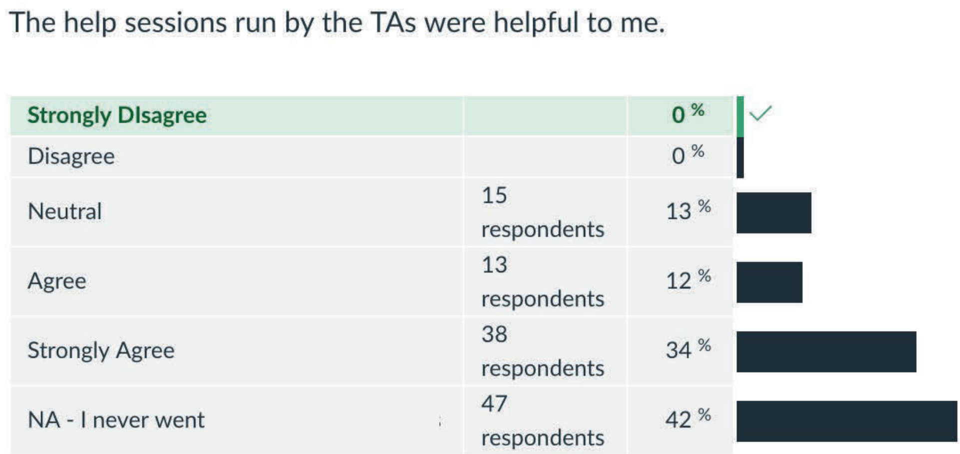

Figure 8.2. Student perceptions about how helpful the TA help sessions were in fall 2018. While some students never attended the help sessions, more than half the class did attend them, and many students found them helpful.

Mentoring Impacts on the Student Learning Experience

When mentoring cycles are effective in active learning classes and in departments or colleges that are investing heavily in active learning, the benefits flow through to students. In our classes, we often ask for detailed anonymous feedback from students both midway through the semester and at the end of the semester. This allows us to continuously improve the classes. Here we present both data from those final class feedback surveys, as well as other sources.

We always end our anonymous feedback surveys with an open question asking students if they have any other comments or suggestions. In more recent semesters, as our mentored TA team has developed, we see this reflected in student comments. Here are comments left by students in the final course feedback surveys in various semesters (edited to show the most relevant parts of the comments):

> ". . . Also, tutoring sessions were fantastic and your TAs tremendously helped me with the assignments and this class." [Fall 2018]
>
> ". . . You were lucky to have such an awesome group of TA's, as was I." [Fall 2018]
>
> ". . . The TA's were also super helpful, especially during the assignments." [Fall 2018]
>
> "Dr. Celine, I loved your class so much! It stressed me out right to a healthy level. Basically, it pushed me, but the end goal was possible to achieve. Your TA selection is phenomenal. Brian, Kyle, and Mariah sat down with me for hours individually just to reassure that I understood the concepts. My 1213 class was done outside of UNCC and I was really scared to come into this class, but they helped me every step along the way. I could not have asked for a better semester, thanks!" [Spring 2018]
>
> "I really liked the layout of this class.... The TAs really made this class a lot of fun and I learned a lot during the help sessions." [Spring 2018]

It seems apparent to students that because the TAs have taken the course themselves, they are really invested in helping the students coming into the class after them to learn the material and build their skills. For example, this comment from a student demonstrates how instrumental the TA was in the student's success in the course: "This class was overall a super positive experience for me. I had a 53% in this class in October and e-mailed Brian and he was super helpful and I ended up finishing the semester with a B, which I'm super happy with. More importantly, I feel that I've become a much better programmer than I was at the beginning of this semester." [Fall 2017] One aspect of the mentoring cycle that has developed is that students in our classes see a well-functioning, cohesive team of TAs who appear to be having fun and are really engaged in the entire class experience. This looks different from many other courses where the TAs are simply used for assignment and exam grading. In these cases, the undergraduate students often do not see the TAs at all unless they meet the TA because of a grading complaint. The much more active and visible role of TAs in our course appeals to students so much that at the end of the semester students themselves want to become TAs for the course. It is amusing that in the anonymous feedback survey at the end of the semester, they ask to be considered as TAs for future semesters, as shown by these two comments:

> "If there is a position for TA next semester I would love to help. I think that this class is essential to making programs more efficient. So, having a good understanding of all the new data structure we learned is essential. Brian had a big impact on me from the labs, so I would like to give another student the same impact." [Spring 2018]
>
> "I loved this class!!! I think it's perfect just the way it is. If you ever need a TA please let me know I would love to help!" [Spring 2018]

In addition to the final feedback surveys, we also often give students reflection exercises throughout the semester to ask them to reflect on their own learning. In one of these reflections, we asked students to reflect on what they found most surprised them about the course. One student responded this way:

> "The effectiveness of the teaching assistants. I have been in a fair amount of classes, many of which have had more TAs than this one; but I've never received help from a TA before beyond just telling me what the correct answer is. I am going to mention this in the course evaluation, also. I really had no idea that a class could actually be accentuated by TAs; I always thought it was only beneficial for the TAs and not necessarily for the students in the class."

This response reflects the fact that the experiences many students have with TAs in other classes are quite limited and possibly even negative. Effective TAing can make for a really positive learning experience for students, and when it happens, students are very appreciative.

We have also seen students comment about the positive experiences they can have in their educational journey when they move through a series of flipped active learning classes taught by faculty who have been mentored in our college. While not every student likes the flipped classroom, many of our students really enjoy the active learning that the flipped class enables, and they comment on how the continuity of flipped experiences across multiple classes is positive. For example, one student left this comment in the end of semester feedback:

> "Thank you for making this class in this format. Having this as a similar format to my 1212 (Bruce) and 1213 (Najjar) classes really helped me understand this class better as I was more prepared for the learning curve that comes with this style of teaching." [Fall 2018]

Another student left this comment:

> "I really like the teaching style of this class. As someone who used the flipped learning method throughout high school, it was very easy for me to transition from high school, to Prof. Najjar's class, to yours. I hope I can take your class again in the future." [Fall 2018]

Another student in the Data Structures course in spring 2018 sent this email to the professor of the course in the fall of 2018, asking about the common structure he has seen:

> "I am curious if you and Professor Ramaprasad (or Dr. Cao or Dr. Najjar) collaborate when designing the structure of your courses' curriculums in any way. I feel as though there are many similarities between the structure of Professor Ramaprasad's ITSC 3146's [Networks and Operating Systems] curriculum and the structure of your ITSC 2214's [Data Structures] curriculum.
>
> "I find that the flipped structure (the focus on videos and prep quizzes and the emphasis on coding and understanding over testing) in your class and Professors Cao's [Discrete Structures], Ramaprasad's, and Najjar's [Intro Programming] classes very conducive to learning." [Fall 2018]

This email demonstrates that some students are noticing the common structure and appreciating how it helps their learning. As more of our classes move toward the FFAL structure, we can expect to see more students who find that this structure provides the support they need to have effective learning experiences.

Limitations

The stepping stone mentorship model for teaching an FFAL class allows support and rich learning experiences for students because there is enough people power to sustain a wide variety of activities and help channels. However, the main limitation to this model is the cost. For a large class, having two faculty involved in the first development semester costs the college or department because one of those faculty members could be teaching something else. However, that is a one-semester cost. Having multiple undergraduate TAs involved in such a class each semester does involve resources. The undergrad TAs are paid $10 per hour. If a class has five undergraduate TAs working five to 10 hours per week, this could impose a significant cost to a department. Thus, this model is most useful for really important core classes. The model could still work scaled down to only one graduate TA and one or two undergraduate TAs. Or the undergraduate TAs could be used for fewer hours per week.

Conclusion

In this chapter, we have presented a case study of creating a self-sustaining, empowered TA team for a large, active learning class as well as a more broadly empowered set of faculty engaging together in active learning pedagogy as a community of practice. We have shown how the stepping stone mentoring model works across multiple levels to empower both faculty and students. We end here with the most salient points for using mentoring to promote and support active learning classes:

- Apply pair teaching to newly flipped classes to minimize workload and allow a faculty member to be apprenticed in the flipped, active learning approach.
- Develop a community of practice within your college or department to allow mentorship among faculty at different points in the active learning pedagogy path.
- Impart the active learning teaching philosophy to the TAs in your course and get them on board.
- Empower TAs to bring forward ideas to improve the class experience.
- Promote mentorship by senior TAs or graduate TAs of more junior TAs and explain the importance of continuity.
- Let TAs interact with students in class and in out-of-class help sessions as much as possible, as this leads to good students asking to join the TA team, creating a self-sustaining cycle.

Some of these practices may be more or less practical depending on resource constraints. We have been fortunate to be able to hire a number of undergraduate TAs for our large active learning classes. These TAs may only work 10 hours a week, and they are not expensive. The investment is worth it, especially for those large core courses where the student learning experience has major impacts on program retention. Having support from the department chair and college dean is critical for these endeavors to be funded and be successful.

References

Barker, L. J., McDowell, C., & Kalahar, K. (2009). Exploring factors that influence computer science introductory course students to persist in the major. *ACM SIGCSE Bulletin, 41*(1), 153–157. doi: 10.1145/1539024.1508923

Bean, J. P. (2005). *Nine themes of college student retention. College student retention: Formula for student success*. Westport, CT: Greenwood.

Dweck, C. S. (2008). *Mindset: The new psychology of success*. New York, NY: Random House Digital, Inc.

Frevert, T., Rorrer, A., Davis, D. J., Latulipe, C., Maher, M. L., Cukic, B., . . . Rogelberg, S. (2018). Sustainable educational innovation through engaged pedagogy and organizational change. *IEEE 2018 IEEE Frontiers in Education Conference (FIE)*, 1–5. doi: 10.1109/FIE.2018.8658491

Latulipe, C., MacNeil, S., & Thompson, B. (2018). Evolving a data structures class toward inclusive success. *IEEE 2018 IEEE Frontiers in Education Conference (FIE)*, 1–9. doi: 10.1109/FIE.2018.8659334

Lewis, C. M., Anderson, R. E., & Yasuhara, K. (2016). I don't code all day: Fitting in computer science when the stereotypes don't fit. *ACM Proceedings of the 2016 ACM Conference on International Computing Education Research,* 23–32. doi: 10.1145/2960310.2960332

Lewis, C. M., Yasuhara, K., & Anderson, R. E. (2011). Deciding to major in computer science: a grounded theory of students' self-assessment of ability. *ACM Proceedings of the Seventh International Workshop on Computing Education Research,* 3–10. doi: 10.1145/2016911.2016915

Maher, M. L., Cukic, B., Mays, L., Rogelberg, S., Latulipe, C., Payton, J., . . . Frevert, T. (2016). The connected learner: Engaging faculty to connect computing students to peers, profession and purpose. *IEEE 2016 IEEE Frontiers in Education Conference (FIE),* 1–8. doi: 10.1109/FIE.2016.7757473

Roberts, E. S., Kassianidou, M., & Irani, L. (2002). Encouraging women in computer science. *ACM SIGCSE Bulletin, 34*(2), 84–88. doi: 10.1145/543812.543837