

Using Undergraduate Teaching Assistants in a Small College Environment

Paul E. Dickson
Hampshire College
School of Cognitive Science
893 West St.
Amherst, MA 01002, USA
pdickson@hampshire.edu

ABSTRACT

The use of undergraduate teaching assistants in computer science courses is not new but is primarily thought of as a way to help with large classes in universities and rarely for small classes in small colleges. In this paper we discuss the success we have had over the past 2 years using undergraduate students as teaching assistants for a number of small computer science classes. Our experience has shown that having undergraduates as teaching assistants helps to engage students with the material, creates a more relaxed classroom environment in which students feel more free to ask questions, improves the effectiveness of class time, and improves class quality. We believe that our experiences using undergraduate teaching assistants can be beneficial not only to small colleges but also to large universities.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science EducationComputer Science Education

General Terms

Management, Human Factors

Keywords

Undergraduate Teaching Assistants, Computer Science Education

1. INTRODUCTION

The best way to understand our undergraduate teaching assistant (UTA) program and its organization is to learn about the environment in which we use UTAs. Hampshire College is a small liberal arts college (approximately 1350 students) with few computer science majors and a student body with a tendency toward the arts. All students are required to fulfill 2 advanced learning activities before they

graduate and one of those activities can be serving as a UTA for a course. Hampshire College also has a large number of students who pursue independent studies, which makes the idea of a student acting as a TA not that unusual. It is within this fertile environment that we began experimenting with the use of UTAs.

We first used a UTA in an introductory computer science course on web development. The UTA was a student who volunteered because he needed an advanced learning activity. The experience was so successful that we have used UTAs in 4 of the 7 courses we have taught since then; 2 of those courses have been introductory level and 2 upper level. Of the 3 courses without UTAs, 2 were new courses for which no student was qualified to be a UTA and the third was a seminar with 12 students for which a TA was not appropriate. The benefit to each course varied from marginal improvement to raging success, with the difference in benefit relating more to the effectiveness of how the UTA was used than to anything else.

In this paper we will discuss how our use of UTAs compares with that at other institutions before discussing details of how we have used UTAs. We will describe the use of UTAs for specific courses, discuss what did and did not work, and then discuss the most effective ways of using UTAs. Finally we will describe the benefits to the UTAs and summarize the UTA program's effectiveness.

2. RELATED WORK

For more than 20 years, undergraduates have functioned as teaching assistants (TAs) for courses and in many institutions UTA programs seem to have been started as a cost-cutting measure. The basic idea was that it would be far less expensive to use UTAs for 100+ person introductory computer science courses than to pay for graduate TAs. While many universities have done this, Reges et al. at Stanford University [6] were some of the first to write about their experience. They found that undergraduate students were quite often willing to be TAs for credit, at least initially. The UTAs would attend lectures, hold their own discussion sections, hold office hours, and grade and discuss those grades with students. What they discovered was that not only was it less expensive to use UTAs but also that they did a better job than graduate TAs because they were more familiar with how Stanford courses were taught. Plus, the UTAs tended to be very motivated and this had a positive effect on the students who were infected by their energy and tended to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE'11, March 9–12, 2011, Dallas, Texas, USA.

Copyright 2011 ACM 978-1-4503-0500-6/11/03 ...\$10.00.

put in greater effort because of it. We have also noticed this effect and it is noted in many of the papers listed below.

Stanford continued this program and over the years streamlined their system and quantitated results [7]. They found that the system not only benefited the students but also the UTAs, helping the UTAs' understanding of the material. They found that the rapport between the UTAs and students in the course was beneficial and the UTAs often acted as role models for the students in the class, which is especially beneficial when you consider that over 25% of the UTAs were women. The course with the UTAs also consistently gets among the best ratings on campus for quality of teaching. One concern was having undergraduates grade undergraduates, but they got around this by grading assignments using a qualitative method that was based more on overall functioning of programs than on specifics of code. One real benefit was that they were able to staff an area to help students with the introductory course and to staff it approximately 70 hours per week except during high-demand times when hours were extended. They believe this program to be a real success.

When Reges moved from Stanford to the University of Arizona, he brought the idea of a UTA program with him [5]. The same positive results were found despite a very different student body. Especially of interest is that this program helped foster a sense of community among undergraduates that was not present before the program and that the interpersonal skills that the UTAs developed impressed employers after the students graduated. Reges's work implies that while benefits vary by institution, the inclusion of a UTA program would likely benefit any institution that has one. The UTA programs at both Arizona and Stanford are very structured, with weekly meetings and structured UTA training time, though they do encourage UTAs to find their own teaching style.

Paxton at Montana State University implemented an undergraduate consultant program that functioned similar to UTAs in a lab environment [3] and without lecture duties. These upper-level students got credit to assist in closed and open lab environments and they and the students in the courses were surveyed to determine the effectiveness of the program. The consultants overwhelmingly believed that the program benefited their understanding of material, communication skills, and understanding of professional responsibilities. The students found the consultants to be effective at answering questions and helping them within the environment.

Appalachian State University developed a peer mentoring program [9] with undergraduate mentors who attend lab sessions and who meet one-on-one with students outside of formal class time. These are essentially the same as UTAs who would attend class and hold office hours. This program benefited the learning of both the mentors and the students. Also, the students who were mentored put forth an increased level of effort.

The University at Buffalo, SUNY changed from graduate TAs to UTAs specifically to improve the level of teaching [1]. There had been many bad experiences with graduate TAs who did not relate to students nor understand how the program at Buffalo worked and therefore the university shifted to UTAs who have an understanding of the curriculum. Using UTAs and giving them a lot of leeway paid off as the UTAs made suggestions about assignments, changed discus-

sion sections to lab sessions, and gave extra review sessions. The authors report that UTAs took responsibility for the course and course material and improved the quality of the course. As a result of this, student satisfaction with the UTAs was far higher than what it had been with graduate student TAs. UTAs were found to be extremely beneficial to their program and to the students involved.

The use of UTAs for large introductory classes has become common enough that their use was mentioned in passing by professors at the University of Washington [8], University of California Irvine [2], and University of Rhode Island [4] when describing courses in various publications.

What all of these programs have in common is that they were implemented in large schools where the courses required some type of TA in order to function. In many of these cases, without some type of TA to grade the professor never would have been able to handle the course. While each paper describes the successes of using UTAs, it is always in the context of providing required additional help for the professor. What none of these papers describes is the situation we have at Hampshire College, which is more common to small colleges: a TA is not required for a course to succeed but does provide enormous benefit if one can be found. In a small college the UTA is supplemental help for the students and professor instead of a required component.

3. UTA EXPERIENCES

The 5 courses for which UTAs were used were taught over the course of 4 semesters and are described in terms of the semester in which they occurred. This chronological presentation shows the progression of ideas behind UTA involvement in courses. The Hampshire College computer science courses discussed here all have between 15 and 20 students and meet for 80 minutes, twice a week, in a computer lab.

3.1 Fall 2008

Our first experience with a UTA was when a student volunteered to be one for an introductory web development course that we were introducing. As this was a first for us, we decided that his duties would be to attend every lecture, take half the grading, put class examples up on the web, and hold evening lab hours twice a week. This workload doubled when we added a second section of the course and had him attend every lecture of both sections. We thought that we were getting some help grading and adding someone to move around the room helping out when in-class examples were given. In actuality, we got a great addition to the course.

The UTA did everything that was asked and more. During class he added details to topics that we had not realized had not been covered sufficiently. He also updated our knowledge of some recent web design tools that he was familiar with because of a side job in web design. As many of our UTAs have done since, he responded to student questions and met with students outside of lab hours on a fairly regular basis. What might be his greatest addition was his contribution to the class environment. Many students find themselves loath to ask questions in a computer science course because they feel out of their depth and small class sizes do not necessarily help enough to overcome this. Having a UTA who is one of their peers sitting in the classroom, sighing loudly, and correcting the professor by stating that this specific piece of code has not been used by any self-respecting web designer in 4 years and is obsolete does a lot to relax students and make

them more likely to ask a question. Unlike when a graduate TA asks a question in class or makes a correction, students view a UTA as their peer and when they see the professor react well to a UTA's question/correction/heckling feel more comfortable verbally participating in class themselves.

3.2 Spring 2009

Our second experience with a UTA was for an upper-level computer graphics course that was being taught for the first time. We enlisted the UTA to hold lab hours so that students could get help with coding issues since most of the class was new to the programming language being used for the course. While the UTA generally attended class, he was never invested in the course. He did his best to try to help students when they had questions with the material but his role was really just as programming support. While his participation as a UTA for the course definitely helped students, his overall benefit was limited as a result of a lack of investment caused by poor planning on our part.

3.3 Fall 2009

When we taught the introductory web development course again we decided to actively look for another UTA for the course based on our previous success. The UTA we ended up bringing in had taken the course the previous fall and had shown considerable interest in the subject material though not the highest level of skill. He was a student whom many of his professors would describe as being flaky. He arrived at college intending to be a film major before gaining interest in computer science thanks to the introductory web development course. We note this information about the UTA because the responsibility of being the TA for this course noticeably improved his work as a student in general. On the last day of the semester, each of the 2 sections of the course gave the UTA an ovation for the outstanding job he had done helping them throughout the semester. This was despite the fact that he had initially seemed like a weak UTA prospect.

The specific tasks for the UTA were the same as for the previous year: to attend all lectures, do half the grading, put class examples up on the web, and hold evening lab hours for students. As with the previous TA for this course, he also made himself available for additional help at odd hours when the students needed it, going above and beyond what was required. It was fairly common to walk into the computer lab where classes and lab hours were held and find his explanations of concepts and problem solving on the whiteboards, left from when he had helped students the previous night. Like the previous UTA for the course, he dramatically lowered our workload by fielding a portion of the questions and grading half the assignments but again this turned out to be more of a side benefit.

Our UTA's biggest benefit to the course was the effect on student moral, attitude, and motivation. Every faculty member finds that by the end of the semester, classes tend to run smoother since students have learned what a specific professor's priorities and teaching style are and know how to best work within that system. By having a UTA who knew that system from day one, students were able to pick up both more quickly. We do not know exactly what our UTA told students about us outside of class, but the students definitely got into the swing of things weeks sooner than they normally do. A specific example of this is that we like to

run an informal classroom because we find that it makes students more likely to ask questions. On the first day of class, about 20 minutes after being introduced as the UTA for the course, our UTA heckled part of the lecture saying something like "Paul, I think you gave that explanation better last year". As we explained to the class how the obvious reason for having a UTA was to have their undivided loyalty and respect, students began to visibly relax. The exchange showed them more about our priorities and lecture style in 2 minutes than they would otherwise have picked up in weeks of class.

3.4 Spring 2010

Based on our continuing success with UTAs, we decided to expand their use to two new courses we were offering, an introductory programming course in C and C++ and an advanced web design course. In both cases we had qualified students to be UTAs.

The introductory programming course had 3 UTAs who had been hand-picked for this course. One was a first-year student with a lot of self-taught programming experience but none in C or C++. We encouraged him to TA the course instead of taking it because he would have found the course boring to take but would prove a valuable reference as a UTA. A second year student who had taken 2 courses with us and who is double majoring with theatre brought with her a different perspective that helped many of the diverse students in an introductory course. As a woman she also was a little more approachable for the women in the course. The final UTA was a third-year student who had taken 4 courses with us and is one of the students found on every campus who understands how things work on that campus and can and did help new students acquire that knowledge. Thus we had 4 instructors in the class and a ratio of 1 instructor for every 5 students.

This course had 2 assignments per week and the UTAs were responsible for grading one of them, a task that they rotated through on a schedule of their own devising. The UTAs also were responsible for holding lab hours 3 evenings per week, attending class, and answering student questions by e-mail. How the semester worked out with the 3 UTAs was informative and influenced how we believe UTAs should be used.

All 3 UTAs were competent and determined their schedules and work distribution among themselves. Although they did not grade equal numbers of assignments, they were happy with the distribution and therefore we were as well. They decided that each would hold lab hours on a different day of the week but kept all the scheduled hours open in their own schedules so that if the one holding hours got swamped, the others were available to go to the lab to help so that every student could get personal attention during lab hours.

The real beauty of the 3 of them was their work in class and as student mentors. In this course students follow along with in-class examples, typing them into their own computers. Quite often they have bugs in their code while doing this. In a typical class with no TAs, the professor either needs to stop the entire class to help a student fix a bug or leave that student behind. With the 3 UTAs, when a bug occurred a student could just call out for a UTA for immediate help so that they could go on with the example. This helped both in keeping them up with class but also with

finding bugs in code or unclear points. The UTAs were in a great position to see if the same problems kept showing up for multiple people or if the code was right and a compiler error was occurring. In each case the UTAs would interrupt lecture to get it addressed or to fill in details the instructor had missed. The UTAs were able to float around the room keeping track of what problems were occurring and were able to funnel information back when it was needed. They also enabled in-class exercises to be done more often as 4 of us walking around a room made it easy to quickly address all student questions that came up in such exercises.

The other benefit the UTAs had was that they acted as mentors for some of the students in the course. Many of the students found that there was one of the UTAs who could explain things in a way that made the most sense to them specifically and started asking questions mostly of that UTA. By the end of the semester, each UTA had a couple of students whom they had taken under their wings and given extra help. This course had a number of students who were taking it because of distribution requirements and who came from artistic backgrounds antithetical to programming. They might have passed the class without the UTAs. As it was, these students were able to get extra help from the UTAs and in a couple of cases many extra hours of help. Not only did these students pass the class but in almost all cases they finished it as a good solid beginning programmers, something they never believed possible at the beginning of the semester.

At the end of the semester the students in the course felt that the UTAs had been one of the best parts of how the course was taught.

The final course for which we used UTAs was an advanced web development course. The UTA for the course was the same one as for the introductory web development course the semester before and so more than half the class was used to having him as a TA. The UTA had limited grading responsibility for this course but was an active participant without whom the course could not have been taught. Our background in web design comes mainly from picking it up to teach courses in it based on student demand. Our UTA was genuinely interested in the subject and trying to actively acquire more skills in the area. The UTA built the Linux server for the course and got all of the appropriate modules working so that we could do server-side development. This meant that when something went wrong or a student in the course needed something done to the machine late at night, the UTA could do it since he was the sysadmin for the course server.

This course was based around a large-group final project. Each group decided on the website they wished to build and by the end of the semester some of the lectures were written purely to fill in information gaps needed by one group or another. For some of these areas we gave the UTA the assignment of researching the topics and giving 30-minute lectures on them. This was only partially successful because while the UTA learned the material and has a lot of one-on-one teaching experience, he did not have lecture experience and his talks were not quite up to the standard of lecture we would have preferred. This suggested that it might be best to in general leave lecturing to the faculty member and let the UTAs stick to individual instruction if the UTA is not going to have regular lecture duties. As before, the UTA handled office hours and put up class examples on the web.

Every student in the course had high praise for the UTA and the help they received from him.

4. EFFECTIVE USE OF UTAS

We believe that the most effective way to use UTAs in a small college environment is to have them attend all lectures, hold lab hours, get class example on the web, keep the lecturer informed on student understanding, and help with some of the grading.

The most important aspect is to have the UTAs present in class. Especially in introductory programming courses, UTAs can be extremely effective in circulating around a room helping students debug code during lecture and as the semester goes on acting as mentors for the students. Most students in introductory computer science courses have recently entered college and how the UTAs act around professors and handle college in general does a lot to inform the new students of how college works and how to handle it. The UTAs also can clarify points that the lecturer may have glossed over and call for the lecturer to slow down or add an extra day of class on a subject if they see that the material is just going over the students' heads. The UTAs have the confidence to state that things are going too fast even when the students do not. By having this power the UTAs start to take responsibility for the course and find themselves equally invested in its outcome and the learning of the students. When they are invested, they put in the extra effort to make sure that the lecturer does a good job and that the students do well.

Having the UTAs in class also does a lot to set the tone for a course. A good UTA will ask questions and interact with the lecturer creating a good dialogue, which pulls the course away from the "sage on the stage" format common to most introductory computer science courses. We have found this to be a very successful way to make students more comfortable about talking in class and asking questions.

Lab hours are also essential because they give the students another time to get help. At Hampshire College few students are willing to wake up for morning office hours and few have time in the afternoon to attend office hours because of course schedules. By having lab hours in the evening, UTAs provide answers to questions at night when students actually need help. Also, if a bad assignment makes it through or is unclear, a UTA can decide during lab hours that it is too flawed to be counted and tell students not to worry about it instead of having them stay up all night trying to figure it out while hoping that the lecturer will respond to an e-mail after 9 pm.

Another good use for UTAs is to have them keep class examples up to date on the web. Instead of the lecturer putting a class example up online after class, a TA who is in class can post examples to the web in real time. This cuts down on the professor's workload and helps students who have poor written notes.

The final area for UTA help is in grading. While the ethics of having undergraduate students grade fellow undergraduate students is open to debate, there is some grading that they can be given in good conscience. Specifically, we have found that giving lots of small assignments in an introductory course is useful because it causes students to engage with the material more often, but to do this effectively all of those assignments need to be graded and no professor has the time to grade multiple assignments per week. As long as

the grading for these assignments is more qualitative and in the range of whether or not the student did the assignment, it is reasonable to have a UTA do some of the grading. The professor's time then can be used more effectively on writing better comments on students' code.

UTAs generally have a good grasp of what is useful when learning computer science material as they generally have been introduced to it far more recently than the professor has been. This means that they can easily remember what made the material clear to them; we have found it useful to listen when a UTA says that the explanation is not clear or that a specific method of going over material will not work. Also, UTAs quite often have more time invested in the specific course material than the professor as they often have a passion in the subject area. If encouraged correctly this can lead to more agile course development as the UTA can be asked to update material that the professor may not have had time to update or in some cases even realize is out of date.

5. BENEFITS TO THE UTA

A UTA program not only benefits the professors who make use of UTAs but also the UTAs themselves. Going over material with another student has the obvious advantage to the UTA that you learn something best when you teach it. At the same time UTAs must be able to talk to the students in the course regardless of background differences and this helps to improve the UTAs' communications skills while also building their confidence. We have seen that many of our UTAs who are a little nervous at the start of a semester show great personal gain by the end and have the confidence to speak up more because of having done so successfully for a semester.

The other major benefit to UTAs is that the experience helps develop their sense and understanding of professional responsibilities as was also noted by Paxton [3]. We have a great example of this as one of our UTAs was known for skipping many of his classes and not being the most dedicated of students. He realized that as the UTA he needed to show up because others were counting him and therefore he did. Not only did he show up to all class meetings and lab hours but he also saw the effect that skipping class had on students in it and realized the disservice he had been doing himself by skipping classes. When students who had set up special meeting with him did not show up, it really brought home how one's attendance and keeping of schedules can affect others. Being a UTA matured this particular student in a way that drastically improved his undergraduate career.

A small side benefit to being a UTA was that it has in each case created a situation of more one-on-one faculty-student mentoring because a UTA needs to work closely with a faculty member for the course to function smoothly. This generally leads to more faculty involvement in the student's academic life and to greater student contact with faculty research. Three of our 6 UTAs have gone on to do undergraduate research and 2 of the 3 are working on projects with us.

6. CONCLUSIONS

We have shown how UTAs can benefit courses even in a small college environment. A UTA program can be run without the level of organization used at a Stanford [6, 7] or

Arizona [5] and still be effective. Being a UTA has benefit for the UTA and therefore it is reasonable to give students credit for their efforts. Having a UTA in a course can cut down on professor workload and enable greater numbers of assignments to be given in introductory courses, improving student performance. Although UTAs are not required in order to teach a good quality course at a small college, they can certainly make the task easier and improve the quality of the course. We have every intention of continuing to use UTAs in our future computer science courses. Despite a lack of specific organization or set structure, the system continues to grow and prosper with strong students volunteering to be UTAs without being asked.

7. REFERENCES

- [1] Adrienne Decker, Phil Ventura, and Christopher Egert. Through the looking glass: reflections on using undergraduate teaching assistants in cs1. *SIGCSE Bull.*, 38(1):46–50, 2006.
- [2] David G. Kay. Large introductory computer science classes: strategies for effective course management. In *SIGCSE '98: Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, pages 131–134, New York, NY, USA, 1998. ACM.
- [3] John Paxton. Undergraduate consultation: opportunities and challenges. *J. Comput. Small Coll.*, 21(1):231–238, 2005.
- [4] Joan Peckham, Lisa DiPippo, James Reynolds, Jake Paris, Phil Monte, and Peter Constantinidis. A first course in computer science: the discipline is more than programming. In *CCSC '00: Proceedings of the fifth annual CCSC northeastern conference on The journal of computing in small colleges*, pages 220–227, , USA, 2000. Consortium for Computing Sciences in Colleges.
- [5] Stuart Reges. Using undergraduates as teaching assistants at a state university. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 103–107, New York, NY, USA, 2003. ACM.
- [6] Stuart Reges, John McGrory, and Jeff Smith. The effective use of undergraduates to staff large introductory cs courses. *SIGCSE Bull.*, 20(1):22–25, 1988.
- [7] Eric Roberts, John Lilly, and Bryan Rollins. Using undergraduates as teaching assistants in introductory programming courses: an update on the stanford experience. In *SIGCSE '95: Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education*, pages 48–52, New York, NY, USA, 1995. ACM.
- [8] Marty Stepp, Jessica Miller, and Victoria Kirst. A "cs 1.5" introduction to web programming. In *SIGCSE '09: Proceedings of the 40th ACM technical symposium on Computer science education*, pages 121–125, New York, NY, USA, 2009. ACM.
- [9] Rahman Tashakkori, James T. Wilkes, and Edward G. Pekarek. A systemic mentoring model in computer science. In *ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference*, pages 371–375, New York, NY, USA, 2005. ACM.