

Collaborative Software Engineering Education between College Seniors and Blind High School Students

Collin McMillan¹ and Amanda Rodda-Tyler²

¹Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
cmc@nd.edu

²Illinois School for the Visually Impaired
Jacksonville, IL, USA
Amanda.Rodda-Tyler@illinois.gov

ABSTRACT

We describe a collaborative software engineering course between sighted college students and high school students with visual impairments. We designed the course as a mentorship experience, in which one college student mentor is connected to one high school student mentee. Each pair of students is responsible for a programming project. The students must learn to communicate programming concepts and software designs, to work with colleagues with very different levels of software engineering knowledge, and to overcome problems related to visual accessibility. We have implemented our course in a pilot program with five mentors and five mentees. This paper covers our course design, initial experiences, and recommendations for future implementations.

Categories and Subject Descriptors

D.2 [Software]: Software Engineering

General Terms

Education

Keywords

software engineering education; visual disabilities

1. INTRODUCTION

Blind students face a struggle after graduation of integrating into a workplace composed mainly of sighted persons [7, 4]. In recent years, this struggle in software engineering has been compounded by the proliferation of GUIs and by the dramatic increase in software size, which leads to difficulties in collaboration and navigation. Several researchers, notably Stefik *et al.* [10, 8], have designed and built effective infrastructures and curricula to teach programming to low vision students. While these materials are growing in popularity, what is still missing in many cases are courses that

implement the curricula for students who have an interest in software engineering. Strong class materials exist, but implementations are relatively rare.

At the same time, a bias has been documented among employers against low vision job applicants [4, 1]. The Associated Press reported in 2008 that “In theory those people are covered by the Americans with Disabilities Act, which among its many provisions requires employers to give fair consideration and treatment to visually impaired employees and job applicants. But [Carl Augusto, president of the American Foundation for the Blind] said employers routinely turn down blind applicants without incurring legal sanction. ‘The ADA is a wonderful law, but many employers find a way not to seriously consider blind people,’ he said. ‘They look at themselves and then say, I can’t imagine how a blind person can be a computer programmer. They can’t possibly do it’” [2]. In other words, employer bias persists due to widespread ignorance about how blind persons interact with computers.

In this paper, we propose a course to address both workplace integration training as well as social biases. Our course is a collaborative, community-based learning design [6]. We connect senior computer science students at the University of Notre Dame (UND), to high school students at the Illinois School for the Visually Impaired (ISVI) and the Indiana School for the Blind and Visually Impaired (ISBVI). One college student is paired with one high school student. Then, each pair of students works together to complete a programming project. Learning objectives for the high school students include programming literacy, computer accessibility aptitude, and college experiential training. Learning objectives for the college students include teaching practice in secondary education, and disability access and sensitivity training.

We have implemented this class in the Fall 2015 (northern hemisphere) term, and it is ongoing at the time of writing. Five college students are enrolled at UND, paired with three high school students from the ISVI and two from ISBVI. While the students’ learning experiences have been enormously positive, the experiences have been highly variable, and logistical challenges are a persistent problem. We describe these experiences and challenges, and provide our recommendations for implementation elsewhere. In addition, we invite advice from the ICSE-SEET community.

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.

ICSE SEET ’16 Austin, Texas USA

Copyright 2016 ACM 978-1-4503-4205-6 ...\$15.00.

2. COURSE DESIGN

This section describes the design of our course at UND, ISVI, and ISBVI.

2.1 University of Notre Dame

At UND, the first author offers a three credit hour course entitled “CSE40586 Low Vision Mentorship”, with a discretionary enrollment cap. The course is advertised in April, and all students who attempt to enroll are added to a waiting list. Later (July/August), as students at ISVI and ISBVI sign up, UND students are pulled from the waiting list in FIFO order. Once enrollment is established just prior to the semester start, all UND students undergo criminal background checks prior to any contact with the high schoolers.

In lieu of a lecture period, the professor holds weekly group meetings. In the first four of these meetings, we read and discuss academic papers related to program comprehension in children [3], visual accessibility technologies [8, 9], and general mentorship activities [5]. During this initial paper discussion phase, schedules are coordinated among students, professors, and teachers, to plan two visits by the UND students. In one visit the UND students travel with the professor to ISVI, and in another visit the students and professor travel to ISBVI. In our implementation, we were able to coordinate such that these visits occurred only five days apart. Group meetings after the visits were dedicated to discussion of issues encountered, lessons learned, and ongoing progress reports of followup email communication among the students.

2.1.1 School Visits

The visits took place over two days. On the first day, the professor drove with the students to one of the schools, with an arrival time around 3pm, to align with the end of the school day. Afterwards, the professor and teacher introduced the students and toured the school for orientation and familiarity. Then, the students met in a computer lab to begin work on a programming project. During this first work session, the college students had the objective to approximate the current knowledge level of the high school student. Some high school students had some experience programming in either formal or self-directed learning. Others had little programming knowledge, outside of a general interest in computers. In all cases, the college students needed to be trained on the screen reader technology that the high school students used to access the computers. A key point to emphasize is that each student had a different level of visual ability, and a different level of programming knowledge. The purpose of the initial work session was to gain knowledge about and build trust among the students.

After the initial meeting (and a group dinner), the professor met privately with the UND students to reflect on the initial meeting. The students presented problems they encountered, discussed potential solutions, and started to create learning objectives customized to their mentee. The professor helped in this customization process, with all students able to comment in a collaborative environment. The students were responsible for a set of learning objectives to accomplish the next day. The day ended around 10pm.

The following morning, students and faculty met for breakfast, prior to a three hour work meeting. The high school students were released from usual classes, to ensure sufficient time to be involved in the project. The objectives for

this three hour meeting varied for each mentee, based on the learning objectives that the UND student mentor determined the previous day. For example, a student starting with no programming experience would learn how to write and compile code, console input/output, conditionals, and other programming basics. The professor and students departed after the long work meeting.

2.1.2 Materials Expectations

There are three expectations for written materials. First, the college students are required to maintain email contact with the high school students. The email contact involves two to five email exchanges per week in which the mentor provides small programming problems for the mentee to solve, as well as providing guidance on the solutions. The mentor is responsible for ensuring that the small programming problems meet a set of long term learning objectives. These learning objectives are clarified in the second written material, a Learning Objectives Document. In this document, the students create a schedule for the mentee related to a programming project that the mentor identifies. The mentor must also write a plan for each objective of how the mentor will assess whether the mentee has accomplished the objective. For example, a game project may be broken into a problem on keyboard input, a problem on audio/video output, and a problem on causing keyboard input to affect audio/video output. Learning progress may be measured in terms of time taken and help required to complete a task. Another useful measure was for the mentor to show the mentee a piece of source code, and ask the mentee to describe what the code does [3]. At the end of the semester, the third written material is a four page summary of lessons learned and descriptions of objectives accomplished.

2.2 Illinois School (ISVI)

The students at the Illinois school enroll in a computing elective course at ISVI. This course meets for one period each day and is taught by a full time professional teacher. The course follows the Quorum curriculum¹ and instruction is in the Quorum language in the SodBeans development environment [8]. While a full discussion of this curriculum is beyond the scope of this paper (it has been discussed and evaluated in existing literature), we do note two key reasons for choosing it. First, it was designed with low vision requirements in mind, and includes strong support for screen reading technology and other accessibility tools. Second, teacher training in the curriculum is available via the EPIQ conference each summer².

The students write a small game as the final semester project for the class. Each day, the students study a programming skill necessary for the game. For example, the students learn about keyboard listeners to handle key strike events, and connect this skill to lessons about object movement in the game world. Both visual and audio interfaces for the game are covered: as objects move on the screen, they also emit sounds to indicate direction and distance (e.g., via pitch and volume). Assignments related to these programming skills lead up to the game project, and the games are entered into a student contest to take place at EPIQ the following summer.

¹<http://quorumlanguage.com/curriculum.php>

²<http://quorumlanguage.com/epiq.php>

2.3 Indiana School (ISBVI)

The Indiana school did not have a full time teacher available to lead a class locally. Therefore, the mentorship program was managed by the Notre Dame students and faculty, with onsite logistical support from ISBVI's administration. Instead of a formal class, the UND students met with the ISBVI mentees between three and five times during the semester under faculty supervision. The number of meetings varied due to student schedules, as well as parent involvement. Meetings involved one three hour session each day, over one or two days. In addition, email communication between mentor and mentee helped the students to communicate problems and progress.

The students collaborated on an engineering project, in which the college student taught the high school student the skills necessary to complete the project. We describe the project we chose in Section 3. Our decision process was guided by the factors we discovered in the first meeting between students (Section 2.1.1) such as experience level and interests. Ultimately, we decided that a project involving both hardware and software would benefit the students, to give the students exposure to both.

3. EXPERIENCES

In this section, we discuss our experiences planning and implementing the first semester of our course.

3.1 Planning/Administration Phase

The planning phase of this course required approximately 18 months of communication among UND, ISVI, and IS-



Figure 1: UND and ISVI students during first visit.



Figure 2: UND and ISBVI students during first visit. Paired programming learning and practice were indispensable. Three hour work sessions were very efficient – longer sessions led to high mental fatigue.

BVI. In March 2014, the first author conceived the idea and began to build connections with partner schools. This extended planning phase was crucial, as it was necessary to secure institutional support at three levels at the schools: from the superintendent, principal, and teacher. Our experience was that all levels were extremely supportive of the program. However, a minimum of one calendar year notice was critical – the schools had to allocate teacher time, acquire approval from parents, and perform criminal background checks on the UND students who would participate as mentors. The first author needed to organize travel time and funds to visit the schools prior to implementation, plus external factors created delays such as Illinois state budgetary uncertainty. Due to these and other concerns, we highly advise that first contact between the college and high school personnel occur 18 to 24 months prior to the first planned collaborative course implementation.

3.2 Recruitment

Recruitment took place at all three schools. At UND, the professor announced the course during the Spring 2015 semester and created a wait list, as described in Section 2.1. Demand far outnumbered supply, with nearly 20 CSE seniors on the wait list, out of 78 eligible seniors. At ISVI, the teacher advertised the course at the school as math elective. Because of graduation requirements and class load restrictions, only students who had completed requirements for graduation were eligible for the class. At ISBVI, the school administration identified two students who had an interest in computer science, and contacted the parents of those students during the Summer of 2015. With parent approval, the students could be placed with UND mentors.

3.3 School Visits

The UND visits to the schools were indispensable in the learning process. We found that these visits should be conducted at the earliest possible opportunity. Email communication prior to the meetings was possible, but the trust developed during the visits was not possible to reproduce electronically in our experience. In addition, we encountered an intense learning curve as the UND students learned to assist the high school students through the accessibility technologies. At ISBVI, the UND students taught the ISBVI students to write programs in C in Linux, using GCC and related technologies. The idea was that these could be taught at the command line, which would avoid accessibility problems with complex GUIs in IDEs. Unfortunately, the configuration of the screen reader on the school Windows computers was not compatible with the SSH terminal access programs³ These difficulties could only reasonably be managed on site, and solving them involved several important learning objectives for the UND students related to accessibility technology. Once this learning curve was solved, computer science training for the high schools was much more straightforward.

The visit to ISVI was closely coordinated with the teacher. The UND visit was aligned with the Quorum curriculum in the class taught by the teacher. The ISVI students had worksheets related to that week's goals in the curriculum, and the UND students helped the ISVI students through

³Eventually we determined that PuTTY with NVDA functioned, even though the students resisted NVDA. Students preferred JAWS, but the cost for that screen reader was prohibitive.

those goals to complete the worksheet. In the introductory work meeting on the first day of the visit, the students worked together on an assignment that the ISVI students had learned before. On the second day, they completed assignments that were new material for the ISVI students.

The hands-on teaching experience was extremely valuable for the UND students. From a software engineering perspective, the UND students were required to practice explaining complex software engineering concepts in a simplified form for the UND students. The students have reported strong growth in two key areas. First, the students have reported improving their ability to verbalize software concepts that “just make sense” to themselves. The concept of function calls was highlighted multiple times as an area that was particularly difficult to explain, even though the UND seniors understood it well. A second area of reported improved understanding is sensitivity to disability issues in the software engineering career path. It is our sincere hope that this improved understanding will spread to future places of employment of the UND students, to help reduce possible biases against persons with a disability status.

3.4 ISBVI Projects

The project that we identified for the ISBVI students and their mentors is to build a high altitude balloon. The students would learn to write a program for a telemetry computer on the balloon. They would also learn to physically construct the computer and balloon apparatus with the help of the UND students. We consider this project achievable, if quite ambitious, for one semester. We use a Raspberry Pi computer with the Raspbian operating system for the telemetry computer, with an attached u-blox 7 GPS USB device. We then preloaded the computer with libgps. The program that the students write records the GPS location every 30 seconds. The learning objectives include: 1) basic C syntax and concepts, 2) file input/output for the log, and 3) concepts behind libraries and APIs by using libgps to obtain the GPS location from the device. The students would also need to learn the supporting technologies such as Linux. A majority of the software concepts were taught through email communication. Two followup meetings are planned to eventually build and launch the balloon apparatus. The hardware device plan is relatively simple: a small styrofoam box will contain the telemetry computer and battery, to be attached to a parachute and balloon via a 10 foot nylon cable. The balloon is a five foot latex design, inflated to three feet at ground level with helium. The helium was obtained through UND procurement.

3.5 Distance Learning

The distance learning component of the course has been valuable, but extremely variable. The high school students have varying levels of interest, maturity, experience, and parent involvement. In two cases, a very high level of contact was maintained between the students, with seven to ten substantive email exchanges per week. In one different case, a moderate level of contact occurred, with between three and five substantive email exchanges per week. Finally, and unfortunately, in two cases, almost no email communication was possible. In our view, the only remedy for this situation is to conduct further on-site visits for those students. These students seemed to struggle with maturity and motivation, even though during the on-site visits, the students

were responsive and learned very rapidly. It is our objective to avoid leaving these students behind. Unfortunately, on-site visits are not possible to coordinate on a weekly basis as desired, due to numerous schedule constraints.

Nevertheless, for the students who communicated readily via email, it was possible to teach computing concepts remotely. For the ISVI students, the UND mentors provided support for the teacher in the context of the Quorum curriculum. As the teacher illustrated concepts in class, the mentors could provide context-specific support for the projects that the students built. Note that the UND students’ time was in addition to the normal time that the teacher could provide; it was not a replacement for teacher time. Therefore, it was possible for the UND students to help the ISVI students in greater detail on specific problems.

4. CONCLUSION

We have described our attempts at increasing participation in Software Engineering of the underrepresented group of persons with visual impairments. Our attempt can be summarized as direct training of the visually impaired students, combined with community learning to increase sensitivity and knowledge of disabilities in the computing field. Our first implementation has had success in on-site paired learning, as well as variable success and difficulty in distance learning. Distance learning is necessary given the hundreds of miles between the schools. Future efforts will improve our procedures in this area.

Acknowledgements

The authors strongly thank and acknowledge the faculty and administration at ISVI and ISBVI. Special gratitude is reserved for Aimee Vieth and Serena Preston at ISVI, and Jay Wilson and James Durst at ISBVI for their advice and invaluable logistical support. This work is supported by the National Science Foundation CAREER Award under Grant No. CCF-1452959 and does not necessarily reflect the opinions of the sponsor.

5. REFERENCES

- [1] S. Alexander. Blind programmers face an uncertain future. In *ComputerWorld*, November 1998.
- [2] D. Crary. Employer bias thwarts many blind workers. In *Associated Press*, May 2008.
- [3] D. M. Kurland and R. D. Pea. Children’s mental models of recursive logo programs. *Journal of Educational Computing Research*, 1(2):235–243, 1985.
- [4] B. O’Day. Employment barriers for people with visual impairments. *Journal of Visual Impairment & Blindness*, 93(10), 1999.
- [5] T. A. Scandura. Mentorship and career mobility: An empirical investigation. *Journal of organizational behavior*, 13(2):169–174, 1992.
- [6] R. Shumer. Community-based learning: Humanizing education. *Journal of adolescence*, 17(4):357–367, 1994.
- [7] T. Smith. Diversity and disability: exploring the experiences of vision impaired people in the workplace. *Equal Opportunities International*, 21(8):59–72, 2002.
- [8] A. Stefik, A. Haywood, S. Mansoor, B. Dunda, and D. Garcia. Sodbeans. In *ICPC’09*, pages 293–294, 2009.
- [9] A. Stefik and S. Siebert. An empirical investigation into programming language syntax. *Trans. Comput. Educ.*, 13(4):19:1–19:40, Nov. 2013.
- [10] A. M. Stefik, C. Hundhausen, and D. Smith. On the design of an educational infrastructure for the blind and visually impaired in computer science. In *SIGCSE’11*, pages 571–576, 2011.