The Apprenticeship Project for educating a STEM-based workforce and the Dissemination of Learning Technology

Richard Alterman, Maria Altebarmakian, Kendall Harsch
Computer Science
Brandeis University
Waltham, MA 02454
Email: ralterma@brandeis.edu

***Abstract*—An apprenticeship community is an approach to linking two very hard problems: dissemination of technology-mediated learning methods and improvements to the preparation of undergraduates for a STEM workforce. A standard computer science curriculum based on knowledge and skill acquisition augmented with a project-based student community of design/builders is a significant addition to current methods for educating students and preparing them for a STEM-based workforce. An ecosystem can be developed that leverages the work of undergraduates as a mechanism to face-lift the technology-base of the classroom. Custom-building education platforms is a necessary and significant complementary approach to generic platforms as a method for propagating technology-mediated pedagogy. This project creates a how-to model for converting institutional need into opportunity for a community of learners.**

I. Introduction

Skill at computation is a valuable tool for any number of fields ranging from the sciences to the social sciences to the humanities and arts. For this reason, many computer science students have strong interests in other fields and are naturally attracted to application-oriented problems. Students of this sort need to be familiar with all aspects in the development of technology, from its purpose and function in a work situation to its eventual implementation; learning to code is only a part of what hybrids must learn.

The claim of this paper is that “hybrid” undergraduates students who are combining computer science with another field of study are the ideal recruits for evolving the technology-base for undergraduate learning institutions. These students are not just programmers. The natural way to educate them is by organizing them into a community and engaging them in projects to develop technology from beginning to end. The idea of the apprenticeship project (AP) is to test whether hybrid undergraduates could work together as a part of a community to custom-build usable educational technology.

This paper presents a one-year study of an undergraduate apprenticeship community. Ten undergraduates were recruited and they were supervised by two PhD students and a faculty advisor. The apprentices built four platforms during the fall semester. Each of the platforms was successfully deployed in the spring. The data presented evaluates the quality of the technology that was built and the experiences of the students and client instructors alike.

The apprenticeship community was set up within the computer science department. As an apprentice, the hybrid student worked with a client instructor, building an educational platform for an individual course. The apprentices were undergraduates in their sophomore, junior, or senior year. The least experienced students were paired with each other on a project, which was mentored by more senior students and a PhD student in computer science. This teaching method combines apprenticeship [1], [2] with project-based learning [3], [4], both of which have many adherents in the learning sciences and computer-supported collaborative learning communities.

The apprentice-built technology supported peer interaction and/or collaboration. It provided opportunities for students in the targeted class to collaborate with one another, either in preparation for an in-class activity, as a reflection exercise after a lecture, or as a platform for students to do homework. As the apprentices designed and built the technology, they learned about some of the current best practices for education and learning as understood by research communities like the learning sciences and computer-supported cooperative learning.

II. Background

The learning the apprentices do is community-centered, practical, constructionist, apprenticeship-based, collaborative, and project-based.

a) Learning community: The concept of learning communities “addresses the needs for students to deal with complex issues, figure things out for themselves, communicate and work with people from diverse backgrounds and views, and share what they learn with others” [5]. Within their learning community, the apprentices have a shared objective, which is to learn how to design and build educational platforms. The learning community is engaged in a public service: they are building educational technology that improves the infrastructure of the university and supports individual faculty in keeping pace with the changing technical landscape of

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

education. Prior studies have shown that projects with idealistic elements like this improve thinking skills and leadership abilities [6], while increasing student gains in moral reasoning [7].

b) Custom-building technology.: The apprenticeship project is a practical approach to solving the problems that emerge from an overdependence on generic learning platforms. A generic learning platform has significant value, but as the gap between what the application does and what a particular class needs increases, the utility of the generic application decreases. This problems is compounded for instructors with low technical skills. Rather than leaving it to the instructor to adapt a generic platform to the needs of her/his course, the AP approach was to custom-build a platform for a particular class or set of classes: the apprentices built a platform that fits the requirements of a specific instructor and his/her class. The apprentices could work more closely with instructors with low technical skills, providing personal help in learning how to use the technology. Thus, custom-building education platforms can be a significant complementary approach to generic platforms as a method for propagating technology-mediated pedagogy.

c) Knowledge building community: The apprentices are part of a knowledge building community [8]. As a result of their work, knowledge within the community of how to design/build a technology-mediated learning activity advances. The apprentices improve each other's ideas. They are learning how to design and how to translate designs into working systems. They engage in discourses that support and enable the creation of new knowledge, reach common understandings, and expand their common base of well-understood designs and codes. The PhD students and the technical advisor provide authoritative information, but the work of the community requires the apprentices put this authoritative information into practice, which requires critical thinking.

d) Project-based: Each project had many elements: the domain content of the targeted class, the computer scientist's task to build the platform, the application of design methods to cognitively engineer the use of the technology, the management of the project, and the social skills to work with a client. Projects were chosen to match the other academic interests of the apprentices; for example, a student double majoring in linguistics and computer science worked on a platform for a foreign language course. Each apprentice participated in all parts of the project. But the apprentices also gravitated towards different parts of the projects; some students preferred the interactions with the client instructors, and other students preferred the system building. To some extent the inner working of each project reflected practice in the commercial world: a working computer scientist may have a job that emphasizes different mixes of these skills.

In general, there are many advantages of project-based learning [3], [4]. The projects connected the students' learning to their everyday experiences, thus providing motivation and context for their learning, while making the knowledge and skills that were acquired meaningful. It enabled the students to develop skill and knowledge in more depth than what they would attain from other kinds of learning activities like doing homework at the end of a chapter. The design artifacts that the apprentices produced made the progress and understanding of each team visible within and across teams, and thereby it became a medium for sharing work, ideas, and feedback.

e) Apprenticeship community: In the apprenticeship community, students engaged in knowledge building and sharing through a collaboration that was both 'vertical' (with "experts" and "novices") and 'horizontal' (with "peers").

In general, "Apprenticeship can be characterised (sic) as a mostly non-didactic way of teaching and learning, grounded in a local context and dependent on participation of the learner in work-related activities; the acquisition of skills during an apprenticeship involves, among others, social participation and interaction, observation and imitation, and engagement through the senses with tools and context" [9].

Apprentice-based learning has adapted to teaching the kind of cognitive skills that are learned in traditional school-based institutions [1], [2]; it is seen as an alternative to standard instructional approaches to teaching skills in a wide variety of technology-oriented fields. Modeling, coaching, scaffolding, articulation, reflection, and exploration are significant elements to the how of cognitive apprenticeship. More skillful members model how to do things. They can also coach more junior members of the community and scaffold tasks. In apprentice-based learning, novices have the opportunity to explore how to do things and are encouraged to articulate and reflect upon their activities.

Each of these six elements to the cognitive apprenticeship model of learning are in play in the apprenticeship community developed in this study. The faculty advisor, the PhD students, and the design mentors were all more skillful at key elements of the work the apprentices did, and they actively coached and scaffolded the activities of the novices. While working on the project the apprentices had ample opportunity to explore, articulate, and reflect.

III. One-Year Study

The community was organized to emphasize peer relations and mentorship among the students. Pairs of student apprentices worked with a client instructor, custom-building educational platforms. Each novice pair was mentored by another undergraduate who was an experienced designer (a design mentor), one of two PhD students, and the computer science faculty member who led the community and was the technical advisor. Before the fall semester began, the faculty advisor and PhD students had an initial meeting with the four client instructors that had been identified. During this meeting, which lasted about an hour, a goal and a basic learning activity to do on a platform was identified for each of the targeted classes.

In total, the community was composed of 10 undergraduates (8 apprentices and 2 design mentors), two PhD students, and a faculty advisor. There were an equal number of male and female undergraduates. The design mentors took an earlier course from the PI and were excellent designers in that class. The two design mentors were seniors. Amongst the apprentices, there were two seniors, two juniors, and four sophomores. All the apprentices had completed at least one semester of programming in an introductory Java programming class.

The apprentices worked on projects for four different instructors. None of the client instructors were in the sciences. Each of the projects was a challenging design-build project, especially considering that 6 out of 8 of the apprentices were sophomores or juniors. Each of the platforms provided support for a different kind of learning activity. The platforms were built using Ruby on Rails (RoR). None of the apprentices or design mentors had prior experience with RoR. Less than half of the apprentices had taken data structures by the time their year as an apprentice began.

The learning the apprentices did depended on peer relations and cooperation, not competition. The apprentices learned together, collaborating in a zone of proximal development [10]. There were differences in skill sets but their skill sets approximated one another sufficient for making useful collaboration a possibility.

During the fall semester, the apprentices met once a week early in the morning for two hours. Coffee, bagels, and cream cheese were provided. As they arrived, the community socialized while eating bagels, discussing project related and non-project related topics.

Early in the fall semester, the PhD students taught RoR to the apprentices, but after the first month the meeting was more in the style of a workshop or studio class, with the apprentices working on their projects. While the students were learning RoR, they also did some simple design problems that made them think about communication and collaborative learning in online environments. The assignments required the students do roughly 2 hours of work each week outside of the meeting. One task given to the apprentices during this period was to design a T-shirt for the apprenticeship community.

By the second month, each project was assigned a pair of apprentices, a design mentor, and a PhD supervisor. As the apprentices designed the technology, they learned about some of the current best practices for education and learning. As the semester progressed, the apprentices worked out a design, regularly met with their client instructor, and implemented the platform. During this period apprentices worked more intensively outside of the weekly community meetings. Each platform had one interface for the instructor and another for the students in the class. User manuals were also produced. In the spring, when the platforms were in use, the apprentices demoed the platform for the targeted class and then supported its use throughout the semester.

During the design phase, the apprentices first produced a paper prototype of their design. They prepared a few slides that described their assumptions about the requirements of the platform and included a few slides that showed their paper prototype; thusly prepared, they met with their client instructor to make sure their assumptions and prototype were on the right track and ask any questions they had. The PhD and design mentors for the project were also in attendance at this meeting but did not lead. The first design meeting with the client resulted in some revisions to the design. During a second meeting with the client instructor, the revised design was confirmed. Only after confirmation of the design did the apprentices begin coding. This design-first approach to building the platforms is the standard methodology taught in courses on interface design [11].

Prior to their apprenticeship, in an introductory programming course, the students systematically learned to code in Java, gradually doing increasingly more complex programming assignments. The students were required to work alone and were also told not to search on the Internet for hints or code to copy. The setup for learning RoR was completely different. The students were encouraged to work together and use the Internet to support their learning. When it came time to build their platform, they worked in pairs in a studio-like environment [12], [13]. During this workshop period, each pair would be working on their code but also talking across pairs about problems that emerged; they all had access via Github to the code other pairs had written. The PhD mentors were there to help out when needed. When one pair figured out how to do something in RoR they would proudly announce their accomplishment to the rest of the community. Rather than learning the RoR language systematically, their learning was task-oriented: learn what you need to know to complete the task.

At the end of the spring, feedback on the platforms was gathered from the client instructors, with the goal to identify minor changes and improvements that could be made to each platform. During the summer, two of the apprentices were given an internship. The interns used the feedback from the client instructors to update the platforms that were to be reused and/or tweak existing platforms for use in other courses.

IV. Evaluation

The applications that the apprentices built recorded some of the activity on each site. Survey questionnaires and interviews were used to collect additional data. The PhD mentors also kept assessment journals, and regularly evaluated the code the apprentices produced.

There are two major questions that will be evaluated in this paper. How well did the apprentice-built learning platforms function? If the apprentices cannot build useful and functioning technology, the projected ecosystem will not work. Apprentice learning is the other critical feature of the project to assess. How well did the community serve as a basis for learning?

A. Value of Apprentice-Built Technology

All the instructors chose to use the technology in the spring semester. All four platforms built by the apprentices were successfully used in the spring. There were very few bugs and all of them were easily fixed. All of the client instructors expressed great satisfaction with the technology in their class and an interest in re-using it in future semesters.

The instructors were greatly appreciative of how well it fit the needs of their class and the assistance they received from the apprentices.

1) Learning the methods of a digital historian: One project was being done with an instructor in African-American Studies. The platform was used in two of her classes. Students were learning the methodology of a digital historian. The students in her class were given a task that involved finding some primary sources online. Their task was to explore available online content, judiciously choosing accurate sources of information, documenting and defending their decision-making as they discovered relevant materials. The platform the apprentices built helped the students construct a narrative that explained their decision-making and recorded their search path as they explored online resources. The instructor reviewed the recorded transcripts and used them to provide better feedback during lecture.

This platform was used in two different classes; one class had 11 students and the other class 12. Between the two classes there were 14 assignments where the students used the platform. The average length in a search path was 4.5 for one class and 6.4 for the other.

2) Building arguments about ethical questions in journalism: In a second project, the apprentices built a platform that supported the collaborative discussion of cases involving questions of ethics in journalism. The instructor was a self-described Luddite. The case studies the students discussed were real world problems, e.g., what were the ethical issues of reporting the name of a suspect in an on-campus date rape situation. Students posted drafts of their arguments and then commented on each other's drafts. The technology the apprentices built converted a standard class into a blended one: in preparation for in-class group problem solving and discussion, the students worked together online to explore various arguments on ethical issues.

This platform was used in one class for three assignments; there were 24 students in the class. On average the students generated a little over 3 comments per drafted argument. On average each post was read roughly 40 times.

3) Post-class reflections: A third project was for a course taught at the Heller School of Social Policy for a mixed class of MBA and MD students from Brandeis and Tufts universities. The platform the apprentices built modified the standard teaching method of Thoughts & Questions (T&Q) (Novak et al., 1999). T&Q works as follows: the students are given some reading and are provided prompts about things to think about. The students respond to the prompts with their thoughts and questions about the assigned material. The instructor reviews the students' responses, using it as a way to focus the lecture on issues that emerge for the students. For the situation the apprentices designed for, the instructor met with a class of MBAs and MDs once a week for several hours; the class was on strategic planning. The instructor used slides during class. After class, the slides were posted to a website where students could respond to prompts from the instructor about the lecture in which they just participated. In this way, the students could do a quick review of a lecture and delve into the material more deeply while also providing the instructor with valuable information about the status of the students' understanding. When appropriate, the instructor could choose to review some of the material or the students' thoughts/questions in the next lecture.

This platform was used in one class. There were 7 assignments and 25 students. Over the course of the semester the students produced 561 different reflections.

4) Building oral competency skills in German: The fourth project was for a German Film class taught in German; the students were at an intermediate level of oral proficiency. As a way for students to practice their oral language skills, they were asked to respond to discussion questions regarding German films. For each discussion question, the students were asked to submit German vocabulary words and a few sentences of notes aimed to focus their response. They also recorded an oral response to the question. All of this information was uploaded to the platform built by the apprentices. Students in the German Film class were able to give and receive constructive criticism from their classmates. This exercise improved the in-class (in German) discussions regarding the films because students had time to develop their ideas and enhance their vocabulary. The online work of all the students also served as a resource when the students were asked to complete a term project discussing topics relevant to the films they watched over the semester.

They were 15 students and 11 assignments in this class. Because the students were graded on their 8 best assignment submissions, participation tailed off towards the end of the semester. As required, each student produced 5 unique vocabulary words for each assignment. Data on how much each student accessed the submissions of other students was not available. The instructor was greatly pleased with the platform and wanted to reuse it for other classes in the future; other language instructors also expressed an interest in modifications of this platform.

5) Surveying the instructors: At the end of the spring semester, each of the client instructors was surveyed. A sampling of comments from the instructors is shown below:

- **Taught students to collaborate:** "The writing in both classes has improved more this semester than in previous years when I have taught both courses. The experience taught them that it is not cheating to think in a collaborative way, to profit from sharing ideas."
- **Prepared instructor for lecture:** "...enabled me to connect with student questions and use them to advance class discussion."
- **Improved discussion in class:**
  - "The technology we used created more transparency and facilitated productive discussion on routes to accurate information through online resources."
  - "The benefits from these out-of class work translated into higher in-class participation in discussions, and often in higher quality of discussions that were set up with a prior online assignment using this software."

– "The students were much more engaged and insightful than I anticipated."
– "...found it particularly wonderful that this tool seemed to help shy students to overcome some of their hesitations of talking in class."

- **Availability of apprentices and sustainability:** " ...offered to implement these minor changes and to make the software available to me again in the coming academic year (Fall 2016 & hopefully for many semesters beyond that since it is such a user-friendly and efficient teaching & learning tool)."

B. Assessment of apprentice learning

1) The apprentices learned to code in Ruby on Rails.: During their interviews, many apprentices commented on the value of this approach to learning a programming language in contrast to the traditional format of lecture and individual assignments. Three of the four projects were finished by the end of the fall semester. The fourth project was completed over the intersession break with the help of one of the PhD mentors. One problem for this project was that both apprentices were athletes with overcommitted schedules. At least five of the apprentices used their knowledge of RoR to build platforms after completing their year in the apprenticeship program.

Several apprentices commented on the differences between the community and the traditional lecture style of learning, for example:

- "It was so different from other computer science classes. I just feel like the other classes, it's interesting to learn things, but this felt more worthwhile."
- "computer science I think you need to learn how to work with a group ...if you're going to work somewhere then that's going to happen and I think it's best to learn it right now. Not discussing with another partner, you're not allowed to do that in 11 or 12 and I think that's not very helpful because if you get stuck and you get limited amount of help from teachers ...it's either that you figure it out or you get handed the answer."

One student commented on the benefits of articulation and reflection:

- "...I had to verbalize more of my thoughts... it was helpful to think through not just what I want to do but why I want to do it because I need to explain that thought process so it makes sense to someone else... sometimes it was easier to solve a problem even once I said it out loud I was like oh, that's the answer!"

Several students commented on the experience of working on a concrete end-to-end project, where the technology would actually be used:

- "For me, I learn a lot better when I understand the real life applications of something and while, yes, there are clear real life applications to the things we learn in CS classes, having a project that I'm working on the entire time knowing that it's going to be used and how it's going to be used and all these other aspects of it really helped me, personally, learn because it helped me understand what it is I'm doing and why I'm doing this."
- "I think I liked the tangibility of it. Just how at first it seemed like we were just learning abstract concepts but we were applying them and actually making something and we could just see everything we were learning in production and see how it worked which I think just helped me learn a lot better than just studying it from a book."
- "The fact that we knew we were making something solid that would be used was motivating because that's something you can show off."
- "This was a lot more full circle... Usually when I have assignments we'll get partial code and have to complete it or we'll have an assignment to do a program that does one little thing and we're told what that is but this was kind of figure out what you want it to do, figure out how to make it do that, and then make it do it so doing all those things and really starting from scratch was very unique."

There were many comments on the value of peer collaboration during the coding process, ranging for working jointly on problems, learning how to work with a group, and refining and debugging ideas:

- "...having a partner was very helpful. In all aspects of it, trying to figure out problems especially in a language that you just learned, figuring out problems is going to be most helpful when you're working with someone else to bounce ideas off of and kind of just figuring out the best way to do stuff."
- "I think it's way more important when you're able to code with other people because it could be something as minor as your code is a little more efficient or writing-wise it's a little cleaner but you might have an idea of how to do something and they might have a better idea or vice versa and being able to communicate your thoughts out before you actually do it, it saves time with trial and error and oh this might work but let's try the better one first' instead of we'll try both eventually."
- "I think by having a partner who had the same exact programming experience that I did, we were able to help each other out when one of us didn't know what to do and by helping each other, I think that reinforced our knowledge."
- "One of the most important skills that a computer programmer needs is to be able to work with a group and be able to work on a real product and to do research basically, that's really important and I think I basically learned that in this class."

2) Community: During the interviews, all of the apprentices expressed a strong sense of community. They uniformly said they would encourage their friends to apply to the program. Many apprentices commented that they got to know each other better than they would have in a regular class and that this made their learning more enjoyable:

- “I feel like we got to know each other better than we do in the other, especially more coding based, computer science classes ... like when I see people around, I say hi to them and I don’t always have that connection in a normal class”
- “I always think about us sitting around the table, eating bagels, drinking coffee, sharing, collaborating . . . that’s what I think of. Working together and enjoying each other’s company.”
- “The computer science classes that I’m taking, you’re basically in a big lecture hall listening to a teacher, we couldn’t ask a question and have a two-way conversation... the fact that we were in a small table, working around made it very social.”
- “I’m kind of shy so in big intro classes I’m probably going to stick to myself and study for tests and program for those assignments on my own so I wouldn’t feel that inclined to try to get into a community . . . I was really sad to leave and the end of last semester because I felt like we were a fun community and I had a good time.”
- “Yes, we learned technical skills, but we learned them as a community and I liked that aspect. ... we know each other so well and we’re helping each other.”

C. Growth as an individual

Many appreciated that the project required they develop the technology from its initial conception to its eventual deployment; it made them feel more confident in their skills and identity as a computer scientist:

- “. . . if I got an error message and I didn’t know what it meant, I could still pick it apart and figure out which parts are more relevant and which parts aren’t going to help you find an answer. Specifically for Rails, it definitely helped because I ended up having an assignment for an interview . . . it was something that I had to do in 48 hours and having had this at least under my belt, it make me feel like I wasn’t just jumping into the deep end of the pool with no way out. . . On a basic level, I felt more comfortable going into that than I would have otherwise because I knew I had already done something that forced me to go out and research a lot of things on my own and just learn on my own... “For me it’s one of those things where like the more experience you get the more confident you become in the skills that you’ve used and this was another way that I gained more confidence because I used and learned more skills and I used skills that I already had.”
- “It was very helpful in boosting my confidence . . . this was more unique in that the end goal was a lot bigger and a lot harder to get to so once we finally got that to work . . . more satisfaction came from that. I think it tested more skills than a regular assignment would do.”
- “Now, with me applying to internships, I’m able to be like I know how to do this, this, and this’. I might not be the best at it yet but I can at least start.”
- “I got real experience with building something and working with a client and I don’t feel like that’s something that many computer science students get a chance to do outside of internships or something. I feel like I have a lot more experience going into the real world.”
- “The apprenticeship is the number one thing that has made me feel more confident. It has made me feel like an actual programmer and not just a computer science student who has done some programming assignments. And it’s really intimidating being around so many other people who have been coding for a lot longer and they made real things and it wasn’t until the apprenticeship that I felt like I also made something real and I’m at their level almost.”
- “. . . I also learned not only how to do things directly but also how to figure out how to do the things I don’t know how to do . . . ”

V. Discussion

A core issue for the proposed project concerns the sustainability of the apprenticeship community. Unlike a regular class, which is self-contained in both space and time, for the apprenticeship community to work, there must be continuation between the generations of apprentices. All of the sophomore and junior apprentices said they would return the following year to mentor the new recruits to the community. Changing roles from one year to the next presents learning opportunities for both the mentor and the mentee. For the new apprentices the relationship with a peer mentor is more empathetic. For the returning apprentice, doing it all over again is a chance to review, reflect, and practice the skill sets they began to develop in their apprentice year, thereby building confidence and competence. Some kind of alumni factor could eventually come into play.

At the beginning of each year new projects must be identified for the new crop of apprentices to work on. In many cases, the projects will be for completely different kinds of platform-based learning activities. In other cases, the project will be built on prior platforms by adding new functionality for the initially targeted class or as part of a rebuild for a new class. For both of these kinds of projects, the design problems must be of sufficient interest and complexity to be good challenges for the new apprentices: not too hard and not too easy.

For the projects that are continuations of earlier projects, how can the knowledge building of the new apprentices be integrated into the work of earlier generations of apprentices? All the projects have a pair of apprentices working on them. If at least one of the apprentices is a non-senior, the chances are greatly increased of having an undergraduate familiar with the design code of each platform still part of the community. By this means, maintenance and development of a particularly significant piece of technology can continue over a several-year period.

What happens to the projects that will continue to be used, but do not require enough change to warrant a new apprenticeship project? This problem was discussed with Director of Library and Technology Services at Brandeis University. From the perspective of the Director, the apprentice-built projects were prototypes of online learning environments that the university could choose to adopt for wider usage. For those platforms that had special significance and wide appeal, technical services would take responsibility for maintaining and modifying in the future. One advantage of this scheme is the apprenticeship community would become a vanguard for the university to keep pace with the changing landscape of learning technology. New ideas for education technology that emerges from communities Computer Supported Collaborative Learning and the Learning Science could be test driven by the apprenticeship community in projects for single classes. The best of the projects would be adopted by the university.

VI. Concluding Remarks

The apprenticeship community is potentially an engine for upgrading and keeping current the implementation and integration of the new science and technology of learning into the university infrastructure. The impact could be twofold: educating instructors and modernizing the classroom experiences of the students. The community is potentially self-sustaining, with one generation of apprentices mentoring the next. It stands as a community that exists for a subset of hybrid students who are full participants in the computer science program with broad interests in computer science and an application science. Apprentices participate in the practice of building applications at the interface of society and technology, will live side-by-side with the regular computer science curriculum, which emphasizes skill and knowledge acquisition. The two systems for learning and training are complementary to one another. Where a course on data structures will help students attain the basic knowledge of a computer scientist, the apprenticeship community puts these skills and knowledge into practice.

The apprenticeship program provides practical experience for the students, making their learning more meaningful. The program addresses issues of alienation that naturally emerge for traditional competition-based models of individualistic learning. An element of doing service for the university by upgrading its technical infrastructure and the altruism it implies has the potential to positively effect student engagement and enthusiasm for science and technology. In this capacity, the apprenticeship program is an on-campus alternative to off-campus internships, which have been argued can provide students the opportunities to apply knowledge and skill sets and offer opportunities to engage with professionals in the workplace [14]. One potential advantage is that the breadth of experience provided for the apprentices is an anecdote to some of the ills of workplace internships, which too often provide narrow experiences focused on a single, entry-level role within a company.

Acknowledgment

This project was funded by a Teaching Innovation Grant from Brandeis University.

References

[1] A. Collins, J. Brown, and A. Holum, "Cognitive apprenticeship: Making thinking visible," *American Educator*, vol. 15, no. 3, pp. 6–11, 1991.

[2] J. Brown, A. Collins, and S. Newman, "Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics," *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, p. 487, 1989.

[3] J. Krajcik and P. Blumenfeld, "Project-based learning," *The Cambridge handbook of the learning sciences*, pp. 317–334, 2006.

[4] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, "Motivating project-based learning: Sustaining the doing, supporting the learning," *Educational psychologist*, vol. 26, no. 3-4, pp. 369–398, 1991.

[5] K. Bielaczyc and A. Collins, "Learning communities in classrooms: A reconceptualization of educational practice," *Instructional-design theories and models: A new paradigm of instructional theory*, vol. 2, pp. 269–292, 1999.

[6] A. W. Astin, L. J. Vogelgesang, E. K. Ikeda, and J. A. Yee, "How service learning affects students," 2000.

[7] J. A. Boss, "The effect of community service work on the moral development of college ethics students," *Journal of Moral Education*, vol. 23, no. 2, pp. 183–198, 1994.

[8] M. Scardamalia and C. Bereiter, "Knowledge building: Theory, pedagogy, and technology," *Cambridge handbook of the learning sciences*, pp. 97–118, 2006.

[9] G. Gowlland, "Apprenticeship as a model for learning in and through professional practice," in *International handbook of research in professional and practice-based learning*. Springer, 2014, pp. 759–779.

[10] L. S. Vygotsky, *Mind in society*. Cambridge, MA: Harvard University Press, 1980.

[11] Y. Rogers, H. Sharp, and J. Preece, *Interaction Design: Beyond Human Computer Interaction*, 3rd ed. Wiley, 2011.

[12] Y. J. Reimer and S. A. Douglas, "Teaching hci design with the studio approach," *Computer science education*, vol. 13, no. 3, pp. 191–205, 2003.

[13] S. Greenberg, "Embedding a design studio course in a conventional computer science program," *Creativity and HCI: From experience to design in education*, pp. 23–41, 2009.

[14] K. Brush, C. Hall, T. Pinelli, and J. Perry, "Interns and mentors evaluation of workforce knowledge and skills and the perceived importance of these skills in engineering and science careers," in *Proc. of the ASEE Southeast Section Conference, Macon*, 2014.