Massive Increase in Eager TAs: Experiences from Extreme Apprenticeship-based CS1

Arto Vihavainen, Thomas Vikberg, Matti Luukkainen, Jaakko Kurhila

University of Helsinki
Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
Fi-00014 University of Helsinki
{ avihavai, tvikberg, mluukkai, kurhila }@cs.helsinki.fi

Final draft
Originally appeared as:
Vihavainen, A., Vikberg, T., Luukkainen, M., and Kurhila, J. (2013). Massive increase in eager tas: experiences from extreme apprenticeship-based cs1. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, ITiCSE 13, pages 123128, New York, NY, USA. ACM.

Abstract

Incorporating students to participate as teaching assistants in our CS1 as early as during their second semester has started a snowball effect, in which more and more students want to be a part of the experience. We allow students to contribute and take responsibility in a context they see as meaningful for teaching. The students-as-teachers approach means that they are mentored by senior teachers in the actual teaching context, which guarantees enough peer and faculty support for students undertaking the task. A significant percentage of our students (ca. 20%) participate as teachers. This has brought us several benefits: (1) new students are welcomed to the learning community by other students as representatives of the institution, not just student organizations, (2) students understand and undertake the responsibility of being a teacher early, and (3) a massive number of eager TAs.

Category[K.3.2]: **Computers and Education** Computer and Information Science Education
Terms: Design, Experimentation, Human Factors
Keywords: teaching assistants, scaffolding, cognitive apprenticeship, extreme apprenticeship

1 Introduction

In order to break the cycle of being *taught to* – that often starts already before primary education – our early computer science (CS) courses are organized to emphasize doing over passive activities, such as being lectured to. Learning opportunities are made possible through working in labs under the guidance of faculty and teaching assistants. A significant segment of our students (ca. 20%) act as student teaching assistants (TAs) in introductory programming courses (CS1) at a very early stage of their studies, often as early as their second semester.

Working as a TA gives the student valuable experience on technical as well as inter- and intra-personal aspects of programming, e.g. communicating with people with different CS knowledge levels and experiencing the significance of best programming practices, all of which are important skills for a CS student. As a bonus, new students are engaged by peers, which has been shown to help increase the retention rate [14] and create a more enjoyable learning context [6].

As being a TA, regardless of the subject, is essentially about working with people, it requires educational structures that take into consideration the learning environment as a whole. TAs need to work in an environment where their work is being recognized as important, but at the same time, successful performance has to be made possible through sufficient support.

Our approach for organizing teaching opportunities for students links with the Extreme apprenticeship method (XA) that we use to organize our programming courses [16]. XA emphasizes communication between teachers and students during the students' problem solving process, which makes the experience of being a member of the teaching community both beneficial and meaningful for participants. TAs constantly perceive how others solve problems, and are able to hone their own problem-solving skills on the side.

We include acting as a TA in CS1 as a formal part of our degree under the course label "CS1 Coaching", as it has clear similarities with agile software coaching (for details see [18]). CS1 Coaching is not a typical course in the traditional sense as there are no teaching-related lectures or workshops, no division into specific groups, and no summative assessment of the students undertaking the duties of a teaching assistant. The only structure is the *way that we organize our programming courses*; the pedagogy we use emphasizes heavy interaction between every participant, benefits from the use of team-teaching, and embraces TAs. Students undertaking the task work as TAs from day one in a real environment, and learn from and reflect on their own experiences. Enough support is guaranteed by the senior TAs and faculty members, who help the newcomers in an authentic working setting.

In the rest of the paper we first describe the pedagogical motivation behind our approach and its connections to student teaching opportunities. We continue by describing the enrollment process and practises required from a student wishing to be a TA, after which we analyze students' expectations and experiences. Finally, we conclude with a discussion on the results and implications of the use of TAs in our context.

2 Pedagogical background

The introductory computer programming education at the University of Helsinki is based on an apprenticeship-based learning framework [1, 3, 4]. This situative perspective on learning is also present in the way we organize learning opportunities for our TAs. According to the situative perspective "knowing is understood as successful situated participation" [4]. In our context it means that knowing how to be a good TA means successful participation in the teaching community of academic teaching personnel within our institution. For our junior TAs, this is realized through the *legitimate peripheral participation* in the community of practice of our CS1 teaching team.

2.1 Students as Teaching Assistants

There is a long tradition of using students and undergraduate students as TAs [10, 5, 6, 12]. In the literature, undergraduate students as TAs in early CS courses typically enroll in a separate teaching course, that seeks to teach aspects of teaching such as "Grading", "Teaching Styles", "How to Present Material" [11].

In our CS programme, undergraduate and graduate TAs have been used as course assistants from the 1970s (the Department of Computer Science was founded in 1969). Like in several other universities, our TAs have been mainly working as appointed course assistants; they have led exercise groups and study groups, corrected exams, and aided lecturers in other course-related tasks, and have received a salary for their work. TAs have been selected based on their grades, attitude, and content knowledge, with the hope that they are good at teaching as well.

In our XA-specific CS1 context, the TAs have no pre-appointed groups to lead, and no responsibility for a specific student group. The whole teaching team of CS1, e.g. faculty, senior and junior TAs, work in XA labs mentoring students. This is motivated by the situative view of learning stressed by apprenticeship-based learning, which emphasizes that learning activities should take place in the same context as they are practiced [1, 8, 4].

2.2 Legitimate peripheral participation

Lave and Wenger proposed legitimate peripheral participation as an activity that leads to learning by integrating newcomers into a community of practice [8, 20]. Legitimacy indicates that the newcomers are recognized as participants in the community by other members and peripherality that they are not expected to fully contribute to the succesful activities of the community.

An individual student is a peripheral or full member of many communities of practice. Participation in these communities leads to learning, and therefore learning occurs all the time and is not confined within participation in formal education. Attention is put on what the communities of practice are that we wish our students to become members in, and what kind of legitimate peripheral participation should be encouraged.

One of the most influential communities of practice, or rather a constellation of communities of practice [20], inwhich the first year students become members, is the community of CS students at the department. Their participation in this community is legitimated institutionally through their acceptance into the university but also through social activities e.g. in student organizations. Legitimate peripheral participation in the community of students teaches newcomers vital skills for coping with their studies and social skills through contacts with peers. Even if this community can prove central in their success in their studies, it does not necessarily contribute to their professional development. Through this participation students can also learn undesirable skills, such as "how to get easy credits" and "how to bend the university's rules for benefit".

There are factors that hinder students' participation in professional academic communities of practice in higher education institutions. Being subjected to tons of mass lectures for instance, can hardly be seen as an engagement in the practices of professional software engineers, as their practice involves mostly activities such as problem-solving, exploration for new information and discussions with colleagues. Breaking the barrier, and letting students truly become members in the teaching community of our department, is the main educational measure taken in our CS1 Coaching course.

2.3 Extreme Apprenticeship

We use a version of apprenticeship-based education called the Extreme Apprenticeship method (XA) in the programming courses offered at our university [16]. XA is based on contemporary interpretations of apprenticeship education in which the emphasis is on teaching crafts that require abstract thinking [1, 3, 4].

XA relies on two key principles: (1) A craft can only be mastered by practicing it, as long as is necessary; and, (2) bi-directional continuous feedback between the learner and the instructor [7]. The second principle is crucial for the existence of XA; feedback enables continuous improvement of the courses.

As a craft needs to be learned by also practising it, programming starts on the very first day of the CS1 course. The first goal of an XA-based course is to have everyone succeed in getting started with programming. It is realized by providing sufficient support in the labs and in the materials. TAs supporting students must restrain themselves from giving full answers to the exercises. If a student asks for help, the task of a TA is to provide just enough hints so that the student is able to discover the answer to the question herself.

As is typical for apprenticeship-based education, XA emphasizes both modeling and scaffolding in the learning process. In the modeling phase, a student is provided with a conceptual model of the programming process in the form of course material, screen casts and lectures with worked examples [2]. During the scaffolding phase, the students work on programming tasks (i.e. exercises) under scaffolding performed by faculty and students taking the CS1 Coaching course.

2.4 Instructional scaffolding

Providing scaffolding for freshmen computer scientists' learning is the main teaching activity in our XA-based CS1. Scaffolding is a metaphor originally used in developmental psychology to describe how a child learns with the assistance of a parent [21]. It means providing well-timed support to the learner's learning process, so that she can achieve learning objectives that she could not reach on her own. Scaffolding is interlinked with Vygotsky's concept of *zone of proximal development*, in which these competencies are obtained through support of more knowledgeable others [19].

Meyer and Tuner describe scaffolding as an "instructional process in which a teacher supports students cognitively, motivationally, and emotionally in learning while helping them to further develop autonomy" [9]. The task that the learner was previously not capable of doing or the knowledge she was not able to grasp, transforms to something she can understand or do on her own [3].

This implies that scaffolding is a process that dynamically changes all the time and is dependent of the learners previous knowledge and progress. As a results, a crucial aspect of scaffolding is its temporary nature. The support given should be immediately dismantled after it is no longer needed, as prolonged unnecessary support can have negative effects on learning [21].

In our context, a significant part of the scaffolding is performed by TAs, who help students in the computer labs. Scaffolding is also embedded into the learning material, especially the exercises, which guide the students to discover the content knowledge embedded in the learning objectives of the course. Dismantling of the support is implemented into the exercises by progressing into larger and more open-ended assignments, after new topics have first been introduced with smaller exercises.

Providing effective scaffolding requires making difficult pedagogical decisions at a rapid pace. Our TAs learn to scaffold students' learning by actually providing scaffolding. In doing this, their learning is scaffolded by senior members in the teaching team. TAs' teaching activities are monitored and a senior member has the capability to intervene in the scaffolding process, but only if the situation requires it.

3 Implementation

Most of the work in an XA-based course is performed in a computer lab, where students working on the exercises are scaffolded by TAs. Each week brings a new set of exercises and knowledge areas to be mastered, while further honing the skills acquired during the previous weeks.

The pyramid of stakeholders in our CS1 is essential for the effectiveness of the labs. There are (1) *teachers responsible for the course* (tenured teachers also participating in the labs) that are at the top of the pyramid, crafting material and exercises, coordinating and controlling the operation; (2) *senior TAs* (TAs on a payroll) who help in labs and contribute to exercises and the material; (3)

	sr. TAs	**jr. TAs**	**students**	**hours**
2010 fall	14	0	192	658
2011 spring	4	4	85	622
2011 summer	3	18	67	484
2011 fall	16	4	197	886
2012 spring	4	15	93	868
2012 summer	3	13	98	487
2012 fall	8	26	189	793
2013 spring	3	15	-	-

Table 1: Number of TAs in our CS1 courses since fall 2010, and the number of hands-on scaffolding hours provided in the XA labs

junior TAs, who are learning to instruct novice programmers and improving their own programming skills by assisting students in the XA labs; and finally, (4) students of the CS1 course, who provide constant feedback on how well the course material and organization works.

As XA emphasizes the individual effort of a student with continuous interaction between all parties, embedding junior TAs in the environment is only natural. The students who participate in the labs give and receive feedback based on their progress, which helps the teachers and learners to reflect and evolve.

Although there is a distinction between TAs, the terms "junior" and "senior" should not be associated with age but with experience. Junior TAs have limited teaching experience, and are typically in a very early stage of their studies, e.g. in their second semester. Becoming a senior TA can happen during a single semester after a student has shown that she possesses the required content-specific skills and is able to scaffold both junior TAs and course participants.

3.1 Origin of the Approach

When we started applying XA in our CS1 courses in early 2010, TAs were selected for the course by our teaching administration. The TAs were both undergraduate and graduate students with sufficient background in programming.

The initial idea for giving the students an early teaching opportunity came from the students themselves; a student who had just finished our XA-based CS1 approached the teacher in charge of the course, and demanded to be allowed to help in the next CS1 course implementation. After realizing the benefits for the students, especially ones within the software engineering subtrack [18], the faculty welcomed the new student helpers.

Initially, CS1 Coaching was not marketed nor included in the official study plan; we simply accepted students who wanted to participate in teaching CS1. The first iterations with early student TAs had only a few participating junior TAs, see Table 1. However, after a while when students started to realize that they could be a part of the *teaching experience*, more and more students started to sign up.

After the initial experiment, the number of willing students has steadily increased: during the fall 2012 semester we had 26 students as junior TAs and 6 students as senior TAs, helping our 189 CS1 students. Currently, as we have more students willing to participate, we have to perform pre-selection in order to make sure that there are enough students to be scaffolded.

3.2 Enrollment

In order to limit the number of students in the CS1 Coaching course, and make the recruitment process more transparent, the students willing to participate as TAs are required to fill an online application form. The form includes basic questions such as grades and the length of studies, as well as open questions "Why should I be a TA?" and "What do I expect from the teaching experience?"

Currently, when the applicant passes the initial filtering (grades are relatively high and studies progressing rapidly), and she is not beforehand known to the course staff, she is invited to an interview. As most of the students are already known by the permanent course faculty from the time they spent in XA labs learning to program, there is seldom a need to interview the students aspiring to be TAs.

Senior TAs are recruited among the more experienced students from the department who have both good grades and study progress, and have skills in teaching either through performing well as a junior TA, performing well as a "normal" TA in other duties at the department, or through work- or hobby-related experience. Selected senior TAs become employees of the department, and are therefore subject to normal regulations of employment. As we organize CS1 three times each academic year, it has been possible to allocate all of the junior TA applicants; if not right away, then in the next cycle.

3.3 Practice

Students that have enrolled to be a part of the teaching experience are motivated and have shown willingness to improve both as programmers and teachers of novice programmers. Their main task is the scaffolding that makes XA-based programming education possible, i.e. helping students to learn programming better than they would simply on their own. This means that students help novice programmers to make working software, improve their design, and nudge them towards necessary information[18]. An important aspect is that the ideas of the students are to be embraced; TAs are forbidden to impose the their own solutions on the students.

As the lab can be crowded, the scaffolders wear bright safety vests in order to stand out in the lab. This also serves as a legitimation of the presence of novice TAs, as this is a sign to CS1 students that the institution authorizes their involvement as instructors. Further legitimization is provided by giving the TAs genuine possibilities to participate in the teaching team by giving feedback on the progress of the course. Each week the staff responsible for the material reviews feedback that has been received during the week, makes adjustments and introduces new parts and exercises to the material. In order to make the release of new material as successful as possible, TAs are encouraged to work through the exercises and material, pointing out possible mistakes for correction. As TAs complete the programming tasks before they are released to the students, they are well-prepared with the material for the challenges in the coming week.

Each CS1 course starts with a meeting of faculty members and the TAs. During the meeting, the team visits the necessary administrative issues as well as the most important pedagogical practices for teaching CS1. Crucial information has been gathered in a so-called "CS1 TA's manifesto" that everyone (TAs and faculty) sign. The two-page document includes the responsibilities and guidelines, of which the most important ones are:

> You will advise everyone in trouble but only give so much help as is necessary for the students to conclude the solution by themselves. Each member of the teaching team has to circulate in the lab monitoring the students' solutions and progress while giving constructive and positive feedback. Comments are given even if the students do not ask any questions. Important things:
>
> 1. **Style of the code**. The students are gradually being instructed to write code according to Clean Code principles. Good style incorporates the following
>   (a) wrongly indented code is not accepted
>   (b) undescriptive names of variables, methods and classes are not accepted
>   (c) encourage to split long methods into separate methods; the guideline length of a method is approximately 10 rows
> 2. **A working solution is not always enough**. A properly working solution should be refactored towards a clearer and more maintainable solution. The person sitting in the next seat should also understand the solution
> 3. **Do not sit and mind your own business even if it is quiet in the lab**
> 4. **Do not touch the keyboard of the student's computer**

It should be pointed out that the TAs do not have the responsibility of grading the students in CS1. The final validation on whether an exercise is correct or not is performed using an automated assessment system [17], which diminishes the potential peer-pressure in grading. The grades in CS1 are based on the number of successfully completed exercises and a written exam.

3.4 Supporting Junior TAs

While interviewing the applicants for the course, we have noticed that most, if not all, have a good sense of how an instructor is supposed to scaffold students in the CS1 course. This is not surprising since students applying to become TAs have participated in the XA-based CS1 course themselves.

Despite existing first-hand knowledge on the organization and teaching in our CS1, the students participating as TAs receive feedback on their actions both from the students and the faculty members. While the feedback from the students is constant in the XA labs, the guidance from faculty members is more subtle. Faculty members as teachers responsible for the course are naturally also present in the XA labs, performing scaffolding to both the students in CS1, and the TAs. As with the CS1, the TAs are not directly told what to do, but just nudged to reflect on their actions from different perspectives. Students as TAs also perform implicit self-reflection with other TAs while instructing, as well as participate in discussions in an active online chat channel.

During the course the teachers responsible for the course organize biweekly face-to-face meetings where all participants of the coaching community are present. Meetings are typically organized as retrospectives. In the meetings, the responsible teachers inspect and reflect on what has been done during the past two weeks, and bring up good and bad experiences and practices to the awareness of the whole team. The team identifies top good practices and marks them as *to-be-kept*, i.e. they should work also during the following weeks. Top bad practices are marked as *to-be-improved* that deserve special focus during the next few weeks. The goal is to have a few of the to-be-improved pile turned into good practices for the future.

The junior TAs are recognized members of the teaching team. In a way, this is also their first opportunity to participate as teachers in the academic community. Formal recognition from the institution in the form of credits and support and the role as "almost teachers" actually makes them participants in the community of teaching staff. The experience is also a valuable merit (albeit not required) when TAs are being recruited for traditionally organized courses at the department.

4 Evaluation and Results

Between fall 2010 and spring 2013 (inluding the currently ongoing course) we have had a total of 116 participants scaffolding in the XA labs. There have been 93 junior TAs (some served in more than one CS1), and 55 senior TAs, many serving as both. During the seven course instances, we have been able to provide 4798 hours of hands-on scaffolding to a total of 921 students. The retention rates of our CS1 have remained on the same level during the junior TA experience, when compared to the retention rates of our XA courses with only senior TAs, in spite of the fact that we have been able to constantly increase the demand level of our CS1.

4.1 Junior TAs' Expectations

In order to gain insight on the students' expectations from the teaching experience, we analyzed 37 free-form applications from fall 2012. We focused on the question "What do I expect from the teaching experience?", and found three recurring themes:

- 76 % expected **learning opportunities**; honing of programming skills, learning about teaching, gaining deeper understanding on different learning perspectives

> ..[I expect] that helping others in courses that I already have passed will deepen my knowledge on the topic, and gives me confidence in my ability to come up with required knowledge when needed....

- 43 % expected **teaching**; helping, guiding, mentoring, coaching, explaining things in words of one syllable

  > ..[I expect] solving problems together with students. Interesting and surprising questions, that force the teacher to look at a problem from a new perspective..

- 38 % expected **social aspects**; working together, meeting new people, being a part of something

  > ..[I expect] meeting new people, the joy of teaching and learning, challenges and opportunities for improving my knowledge, and new experiences in good company..

Most of the students (76% of the correspondents) were expecting new learning opportunities and improving their own skills. The skill set that the students were hoping to improve varied a lot; most of the students indicated that they would like to become better programmers. A clearly recurring sub-theme was also learning to work better with people.

Although we labeled the second theme teaching, the students that expected teaching (43%) did not expect teaching as a traditional “talking head”. Almost all of the students further explained that they want to help students figure out the solutions themselves, or that they want to work on problems together with the students. This is to be expected as the students themselves have taken our CS1 course where the focus is actual doing, and they have already seen the TAs in action.

The third theme was about social aspects. Some of the students were eager to join to the community of teachers and meet other students that were interested in teaching. The good mood in the XA labs shone through in some of the applications; as the students had received good experiences in the lab as students in CS1, they wanted to continue and bring the same feelings to new students.

4.2 Junior TAs’ Experiences

In addition, we posted a questionnaire to students that have been participating as junior TAs in an XA-based CS1. A total of 46 replies was received. The questionnaire contained Likert scale questions that measured the students’ experiences on the scale 1=’not at all’ to 5=’a lot’ and “cannot say” which is not considered in calculating the mean μ and variance σ^2.

Interpersonal skills scored relatively high:

- I have become more proficient as a member of team of teachers: $\mu = 3.00, \sigma^2 = 1.04$
- I have become more proficient in understanding problems posed by others: $\mu = 3.72, \sigma^2 = 1.03$

- I have become more proficient as an instructor and mentor: $\mu = 3.64, \sigma^2 = 0.66$
- I have become more proficient at recognizing learning styles of others: $\mu = 3.11, \sigma^2 = 1.29$

TAs were also asked for open comments regarding their experience. The comments revealed that the experience was highly valued. Many of the students stated that they now “understand my own learning process better” , and that often “simply listening and being present helps students to find the solution”. Other, more targeted open comments included “learning to cope with chaos” and “learning to be more patient”, as well as references to “learning to switch fast between mental tasks”. The comment that seems to summarize the whole process was that “teaching is hard but wonderful”.

4.3 CS1 Students’ Feedback

Of course, the use of novice TAs can only be justified if it does not have a negative impact on the learning of the students they teach. We have witnessed an overall decrease in drop-out rates and increase in students’ persistence due to our measures taken in CS1. There are many factors contributing to the success of students and a more thorough investigation on the subject can be found in [15].

In order to measure the CS1 participant satisfaction with regards to scaffolding in the XA labs, we have gathered anonymous feedback from the course participants using an online form. Although giving of feedback is not mandatory for the students, we received 189 anonymous feedback answers during 2012.

Overall, the teaching organization and course content is seen as extremely successful, and the course constantly receives top scores in anonymous feedbacks.

Out of the 189 comments, 30 included comments about the labs, and were all positive. Both the junior and senior TAs (as well as faculty) received tons of praise, while 6 comments contained negative remarks. The negative comments concerned with a TA not having prepared for the lab (1), the junior TAs not always being able to help (3) (the comments noted that the senior TAs and faculty did help out though), and the availability of help during busy times (2). The availability and activity of support also received praise, and the satisfaction towards the course is also visible in the number of students that want to become a TA.

5 Conclusions

We started using XA in our CS1 in 2010, and the first junior TAs enrolled to the course in fall 2010. Since fall 2010, we have been able to offer a significant amount of hands-on support for the students striving to be better in the craft of programming.

Involving students in the teaching community as early as possible has been beneficial and fruitful for all parties. The freshmen CS students benefit from a well-organized teaching structure, while the TAs hone their own skills in teaching and programming-related skills. Participating as a junior TA can be seen as a signal of interest towards academic life, and many of the TAs have been selected as senior TAs as well as into research groups.

When considering the expectations of the students, it is clear that the students have **learned** and participated in **teaching**. The most important benefit for the students participating as TAs is the experience that can be used for reflection on upcoming courses as well as in the job market. Students learn much about interpersonal and social aspects that are often otherwise neglected in CS studies.

It is well known that "faculty have been drawn away from undergraduate instruction in order to teach graduate courses and fulfill increasing research demands [...] a great deal of responsibility for undergraduate instruction has been handed over to graduate teaching assistants" [13]. We do not use students in CS1 to save or reallocate resources to "more important" courses. We consider CS1 the most important course of our BSc degree; that is exactly why we want to make it as good an experience as possible.

Deploying "rookie" students as TAs has not deteriorated the learning results when switching from traditional lecture-based CS1 to XA [15]. In fact, the number of TAs has allowed us to significantly increase the amount of support and lab times available for our students. We claim that enough support structures (hierarchy and heavy interaction between all participants) ensures the quality of scaffolding for the actual students of CS1.

References

[1] J. Brown, A. Collins, and P. Duguid. Situated cognition culture of learning. *Educational Researcher*, 18(1):32, 1989.

[2] M. E. Caspersen and J. Bennedsen. Instructional design of a programming course: a learning theoretic approach. In *ICER '07: Proc. of the 3rd international workshop on Computing education research*, 2007.

[3] A. Collins, J. Brown, and A. Holum. Cognitive apprenticeship: Making thinking visible. *American Educator*, 15(3):6–46, 1991.

[4] A. Collins and J. G. Greeno. Situative view of learning. In V. G. Aukrust, editor, *Learning and Cognition*, pages 64–68. Elsevier Science, 2010.

[5] A. Decker, P. Ventura, and C. Egert. Through the looking glass: reflections on using undergraduate teaching assistants in cs1. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, SIGCSE '06, pages 46–50. ACM, 2006.

[6] P. E. Dickson. Using undergraduate teaching assistants in a small college environment. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, pages 75–80. ACM, 2011.

[7] J. Kurhila and A. Vihavainen. Management, structures and tools to scale up personal advising in large programming courses. In *Proceedings of the 2011 conference on Information technology education*, SIGITE '11, pages 3–8. ACM, 2011.

[8] J. Lave and E. Wenger. *Situated Learning: Legitimate Peripheral Participation*. Learning in Doing. Cambridge University Press, 1991.

[9] D. Meyer and J. Turner. Using instructional discourse analysis to study the scaffolding of student self-regulation. *Educ. Psychologist*, 37(1):17–25, 2002.

[10] S. Reges, J. McGrory, and J. Smith. The effective use of undergraduates to staff large introductory cs courses. In *Proceedings of the nineteenth SIGCSE technical symposium on Computer science education*, SIGCSE '88, pages 22–25. ACM, 1988.

[11] E. Roberts, J. Lilly, and B. Rollins. Using undergraduates as teaching assistants in introductory programming courses: an update on the stanford experience. *SIGCSE Bull.*, 27(1):48–52, Mar. 1995.

[12] R. Santa Maria and S. Banerjee. Using undergraduate teaching assistants in introductory computer courses. *Journal of Computing Sciences in Colleges*, 27(6):61–62, June 2012.

[13] D. M. Shannon, D. J. Twale, and M. S. Moore. Ta teaching effectiveness: The impact of training and teaching experience. *The Journal of Higher Education*, 69(4):pp. 440–466, 1998.

[14] C. Stewart-Gardiner. Improving the student success and retention of under achiever students in introductory computer science. *Journal of Computing Sciences in Colleges*, 26(6):16–22, June 2011.

[15] A. Vihavainen, M. Luukkainen, and T. Vikberg. Results from a three-year transition to the extreme apprenticeship method. To appear in *Proceedings of the 13th IEEE International Conference on Advanced Learning Technologies*, July 2013.

[16] A. Vihavainen, M. Paksula, and M. Luukkainen. Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, pages 93–98. ACM, 2011.

[17] A. Vihavainen, T. Vikberg, M. Luukkainen, and M. Pärtel. Scaffolding students' learning using Test My Code. To appear in *Proceedings of the 18th annual joint conference on Innovation and technology in computer science education*, July 2013.

[18] T. Vikberg, A. Vihavainen, M. Luukkainen, and J. Kurhila. Early start in software coaching. To appear in *Proceedings of the 14th International Conference on Agile Software Development*, June 2013.

[19] L. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1978.

[20] E. Wenger. *Communities of Practice: Learning, Meaning, and Identity*. Learning in Doing Series. Cambridge University Press, 1998.

[21] D. Wood, J. S. Bruner, and G. Ross. The role of tutoring in problem solving. *The Journal of Child Psychology and Psychiatry and Allied Disciplines*, 17(2):89–100, 1976.