Lessons Learned on Computer Science Teachers Professional Development

M. Cecilia Martínez
Centro de Investigaciones
FFyH/CONICET, Universidad
Nacional de Córdoba
Córdoba, Argentina
cecimart@gmail.com

Marcos J. Gómez
Logic, Interaction and
Intelligent Systems Group
FAMAF, Universidad Nacional
de Córdoba
Córdoba, Argentina
mgomez4@famaf.unc.edu.ar

Marco Moresi
Computer Science Section
FAMAF, Universidad Nacional
de Córdoba
Córdoba, Argentina
mrc.moresi@gmail.com

Luciana Benotti
Logic, Interaction and
Intelligent Systems Group
FAMAF, Universidad Nacional
de Córdoba/CONICET
Córdoba, Argentina
benotti@famaf.unc.edu.ar

ABSTRACT

This paper describes an introductory Computer Science (CS) Professional Development (PD) course for K-12 teachers in Argentina that integrates pedagogical content knowledge and teacher classroom practice. We analyzed teachers' learning of what CS entails and the implementation of inquiry-based programming lessons in their schools. Based on pre and post teachers surveys and classroom observations, we found that most teachers learned about the CS object of study and about fundamental programming concepts such as conditionals, loops, variables, etc. Teachers were more likely to replicate the same activities they experienced during PD workshops in their classrooms than to produce their own. Teachers who had a previous background on CS provided in-depth explanations of CS concepts to their students while other teachers superficially introduced the content knowledge. We describe PD activities and characteristics that could explain teachers' learning and incorporation of programming lessons. Findings imply that a PD program that integrates pedagogical content knowledge and teachers classroom practice can effectively improve inquiry-based CS teaching, but may be insufficient preparation for teachers with no previous background on CS.

CCS Concepts

•**Social and professional topics → K-12 education;**

ITiCSE '16, July 09-13, 2016, Arequipa, Peru

DOI: http://dx.doi.org/10.1145/2899415.2899460

Keywords

Computer science K-12 outreach, experimental evaluation, Computer Science (CS) Professional Development (PD)

1. INTRODUCTION

As many countries are moving forward in their efforts to introduce Computer Science (CS) massively into the mandatory school curriculum, one topic of debate between academics, policy makers and the whole educational community is who is going to teach CS in schools and how are teachers going to be prepared. Currently, one of the major challenges for teaching CS is the lack of teacher subject knowledge [8].

While most researchers and policymakers agree that teacher certification degrees offered by Universities are the best option to prepare highly qualified teachers [13], some countries are considering training teachers "on the job" (such as the UK, New Zealand, the US and Germany) [14, 7, 11, 4]. These countries face the dilemma to move forward with their CS curriculum reform whit a shortage of CS professionals willing to move into a teaching path. Hence, they opt for in service training or short term Professional Development (PD). In Argentina, for the last five years members of the National Secretary of Science and Productive Innovation together with National Universities have been promoting the teaching of CS, and in particular CS programming, in mandatory education.

Researchers have documented that effective PD programs, that include an explicit focus on the subject matter and analyze student thinking, can promote teacher learning [6]. Requiring teacher active learning through workshops or study groups within a coherent program where learning goals, content and activities are aligned, has significant effects on positively changing teacher learning [6]. When it comes to teacher PD in technology, first hand experiences with technology, combined with observation and analysis of other teachers using technology are effective strategies to change teachers beliefs and practices about technology [15]. While

research on effective PD and on introducing technology into teachers practices is vast, little is known about teacher pedagogical and conceptual learning during CS PD programs. Also, we ignore how teaching practice can change as a result of these programs. With the purpose of studying both teacher CS learning and contributing to promote best CS education PD practices, we designed an exploratory study to analyze how the PD learning experiences impacted teaching practices in CS, and how teachers changed their beliefs about the subject area. We offered a 50 hours introductory course on teaching programming at the Universidad Nacional de Córdoba, in Argentina for K-12 teachers. The course required 10 hours of teaching classroom practice at their schools. Advanced CS college students provided coaching. In this paper, we describe the innovative characteristics of our PD course and show how teachers implemented CS programming lessons learned at our workshops. Our findings can inform the design of effective CS PD programs. In the next section, we describe previous studies that analyzed CS PD events. Then, we present the design of our PD plan supported by theoretical foundations. The section that follows, explains the methodology of the study. In a fifth section, we present our findings. Finally we draw some conclusions and discuss implications.

2. PREVIOUS WORK

The current literature on CS teacher development, whether it is on teacher preparation or in-service training, points out at two major structural issues. First, there is no consensus on what CS education involves and what the minimum requirements should be [8]. Second, it is not clear which higher education institutions should offer teacher preparation programs. CS teachers preparation programs in Argentina and other countries do not have a clear definition of the field and confuse CS with other subject areas such as Technology Education/Educational Technology (TE/ET), Industrial or Instructional Technology (IT), Management Information Systems (MIS), or even the use of computers to support learning in other subject areas (CSTA)[1].

Second, current CS or informatics teachers (in the few schools where the subject is taught) do not necessarily have CS backgrounds with many being from "ICT", which focuses more on training computer users rather than software developers [14, 7, 4]. Most CS teachers are science or math teachers who register on in-service programs, online courses or short PD offers to teach CS [7, 11, 12] .

A third issue that is not particular to our country is that Argentina can not currently cover the demand of programmers and CS engineers the industry requires, thus is it very difficult to attract the workforce towards a traditionally lower pay sector such as education.

Given these structural conditions (lack of coherence among CS preparation programs, current CS teachers with heterogeneous CS backgrounds, lack of CS professionals interested in teaching) many countries have considered preparing teachers through short-term PD. However, there are no enough documented experiences that can provide some information about how these initiatives resulted. Previous work showed that a short training experience in the format of summer workshops could not modify the teaching of CS. Moreover, teachers with no previous knowledge of CS are more likely to drop out of these training courses [4]. During an experience in Los Angeles School District where teachers had two weeks of summer course on Advance Placement in Computer Sciences (APCS) teachers reported successfully engaging their students in introductory CS ideas. Moreover, in a few years, the program doubled the amount of schools offering APCS courses in the district [7], not a minor milestone when many CS courses are closing in the US because so few students register [3]. However, teachers expressed a concern on their ability to teach more advanced APCS concepts and advocated for further training opportunities especially teachers with the least prior CS knowledge [7]. In theses experiences, teachers expressed confidence in launching the course, but reported trying "to stay one step ahead of students" as the course progressed and they had to present more difficult concepts to the class.

Thompson and Bell [14] reported that teachers self reported programming experience and confidence on teaching CS has improved in New Zealand after two years of the CS curriculum implementation in the country. Surveyed teachers took a variety of PD training offered by non governmental organizations. While overall studies seemed to show that teachers gained confidence in CS education, the research literature is not clear on how teachers incorporated what they learned in their classrooms and what relation can be established between PD activities and teaching practices.

In the next section we describe the design of our PD program to explain possible links between PD activities, teacher learning and classroom implementation.

3. A COMPREHENSIVE APPROACH TO TEACHER PD IN COMPUTER SCIENCE

For the last three years we have organized an introductory CS programming teaching PD course for K to 12 educators at the Universidad Nacional de Córdoba in Argentina. In 2015 the National Ministry of Science decided to replicate our course in 10 other national universities in the country in order to engage more public universities in a national effort to increase the number of CS teachers.

We offered a 50 hours long course with classes distributed across one academic semester. 40 of these hours were taught at the University. We hold 4 to 6 hours workshops every three weeks for the teachers. In addition, teachers had to complete 10 hours of "teaching classroom practice" in their own schools delivering CS lessons to their students. That is, teachers had to teach one hour of CS per every 4 hours of PD. Advanced CS college students, who we call "coaches", worked with each teacher individually for 4 extra hours reviewing lesson plans and providing in class support. Coaches help teachers pick the programming platform that teachers prefer and propose them different strategies for focusing their lessons on programming concepts instead of doing it on the selected programming platform.

We require teachers to register with another teacher from their school to reduce teacher isolation in implementing innovations. Research suggests that the most important predictor of teachers using computers in their schools is the computer use of their peers [5]. We offer teachers a vast array of curricular resources including CS inquiry based lessons plans specially developed for this course.

[1]CSTA National Secondary School Computer Science Survey: Comparison of Results from 2005, 2007, 2009, 2011, and 2013 Surveys

Three modules: Robot, Chatbot, and Animations; purposely address fundamental programming concepts such as sequence, event, conditional, cycle, variable, method, parameter, among others. Technological content knowledge is also present in our course as we promote the use of different computer platforms specifically designed to teach programming such as the well known Alice, Code.org tutorials, and our own developed open sources teaching CS platforms: Chatbot which allows programming chat automatons [2], and UNC++Duino, a multi programming language platform for Arduino robot programming [1]. We repeatedly teach the same CS concepts in all platforms, thus focusing on a conceptual approach to learning CS, with the goal that teachers do not become platform dependent. We strongly convey the message that teaching programming is not about using a platform or learning a particular programming language, but about learning CS concepts and computational thinking skills.

Each of the teacher meetings follow an inquiry based approach. In a short first 5 minutes segment of the meeting, we introduce the particularities of the selected platform. Before introducing any CS concept, we give teachers a programming challenge. For example, developing an obstacle dodger robot using the robot programming software, UNC++Duino. We require teachers to solve the challenge in groups of 4 to 5 people promoting teacher collaboration. Coaches walk around the room to assist teachers and prevent anxiety and frustration. Intentionally, we train coaches to avoid giving teachers the solutions but rather to guide them with questions. Based on the literature, we know that most teachers have a tendency to teach with the same teaching strategies they were taught, and that learning pedagogical theories can not compete with what teachers experienced as students.We also know that most teachers did not have previous experiences in learning CS in their schools, thus, we want to create an inquiry based and meaningful first time, first hand programming learning experience [15] that can have impact in their future teaching. In addition, to encourage teachers implementing programming lessons into their classrooms, we invite teachers and their students to an end of the year programming fair at the University. In this fair, students of participant schools bring their programming productions (video games, animations, chat automata or robots). The fair has proven to be an important incentive for teachers and students to bring programming projects into their schools. Teaching teachers through programming challenges allows the integration of CS concepts, inquiry based pedagogy and technology. University coaches support these complex learning process enriched with in classroom practical experience. This format permits preparing teachers in pedagogical technological content knowledge [10] necessary to provide high quality CS teaching.

4. STUDY DESIGN

We wanted to know what teachers with different CS backgrounds were learning in our introductory CS programming teaching PD course offered in 2014 and 2015. Specifically we wanted to document what CS concepts and teaching strategies teachers were selecting to teach their lessons and how these strategies relate to the activities we offered in the PD hours. An exploratory research was best to answer these open ended questions.

4.1 Sample

Our course was open to all primary and secondary school teachers who were willing to teach CS in their classrooms. We advertised it in social networks, the university web page and local news TV channel. Therefore, our sample included self selected teachers who had access to these channels of communication. The course was free of charge for teachers and financed by Google in Education in 2014 and by the National Ministry of Science in 2015. All the requirements to complete the course were informed up front. Google, the National Ministry of Science and National University endorsement provided strong legitimacy for the course.

46 teachers completed the course in 2014 and 60 did in 2015. For several reasons (personal, institutional, and cognitive), about 10% of the registered teachers could not fulfill all the requirements to complete the course. Teachers' background and profile was heterogeneous. Table 1 summarizes the profile of the teachers who registered for our course. Primary school teachers in Argentina have a general preparation in pedagogy and primary school content knowledge. Most primary school teacher preparation programs include one course called "technology" and another course called "audiovisual media". But none include formal CS programming content. Among the secondary schools teachers, 22% of them held a degree in a CS related field such as programming technician, informatics technician, or bachelors degree in CS systems or alike. 20% of the teachers held a tertiary degree on educational technology. Most educational technology programs prepare teachers to integrate ICT in schools, but do not address CS subject matter. The rest of the teachers held degrees in math, language arts, arts, and chemistry education. Many of the teachers who held a degree on educational technology were teaching subjects such as computers labs or informatics. In a previous study [9] we documented that some CS high school courses offered general ICT preparation, most likely because of lack of teachers' CS content and pedagogical knowledge.

Teachers general profile	**%**
Public School Teachers	75%
Secondary school teachers	60%
Primary school teachers	20%
Secondary and tertiary school teachers	20%
Teachers with a degree in CS related fields	22%
Teachers with a degree in educational technology	20%
Teachers teaching technology courses	13%
Teachers teaching CS related courses	25%
Teachers teaching Mathematics	20%

Table 1: Summary of sample profile (N=106)

4.2 Data Collection and Analysis

Teachers completed three online surveys that included 27 open ended and multiple choice questions. The first survey, completed upon registering for the course, gathered teachers information on previous preparation and current position.

After reviewing the literature on variables and conditions that affect teachers appropriation of curriculum and pedagogical innovation; we designed the second and third questionnaires that were used as teachers pre and post survey. The survey gathered information on teaching styles, use of computers, and teachers representations about CS.

All teachers responded to the mandatory first and second questionnaires. However, in spite of emphasizing to teachers that completing the end of the course questionnaire was important, only half of them did so. Besides the questionnaires, the coaches wrote semi structured classroom observations after every lesson they participated. The observations included items such as students and school demographics, CS content knowledge selected for the lesson, description of teachers expertise delivering the content, description of teaching strategies, best moment of the lesson, and things that could have been done differently. In all, we gathered 73 coaches observations for both the 2014 and 2015 editions. Because teachers registered in pairs, some of them also decided to deliver the lessons in pairs, in those cases, the coaches wrote only one observation for both teachers. Teachers questionnaires and coaches observations were the main source of data for this paper.

We created frequency tables for the multiple choice responses and analyzed qualitative data inductively. We identified emerging themes upon reading teachers and coaches discourses and from these themes we constructed analytic categories. 9 members of our team, including the coaches and research assistants, coded coaches observations. To ensure reliability, each observation was coded more than one time by another researcher rather than the coach who wrote the reflection.

Teachers questionnaires were analyzed in a spreadsheet also inductively assigning emerging codes to the teachers responses. We compared pre and post survey responses only among teachers who responded both questionnaires. Peer and multiple coding of data and triangulation of teachers questionnaires with the coaches observation contributed to the validity of our study as we could contrast and compare our emerging themes with different sources of data. As a result of this analysis, we developed broad emerging themes that helped us answers some of our research questions.

5. FINDINGS

In this section we describe our findings with respect to what the teachers learned during the PD training and how they were able to apply it in their classrooms.

5.1 Teachers learning Computer Science

A comparison of pre and and post teacher survey responses showed that teachers changed their definition about the discipline CS and increased the number of CS concepts they reported to know.

Because teacher understanding of the discipline is central to teach CS, we asked teachers with an open ended question in the pre and post survey what CS was for them. We classified teachers responses and grouped answers in 6 emerging themes. Table 2 compares teachers answers.

Emerging theme	Pre-test	Post-test
Imprecise definition of CS	18.5%	0%
Theory in informatics	33%	35%
Computer programming	22%	15%
Information processing	15%	8%
ICT	11%	0%
Discipline to solve problems	0%	38%

Table 2: What teachers think CS is (N=54)

About 30% of the teachers answered in both pre and post survey that CS included the theories, disciplines and sciences that study informatics. About 20% mentioned that CS was related to programming. 18% of teachers gave broad and unclear definitions of CS in the pre survey. For example, teachers mentioned: “CS is about electronics”, “CS is about the computer”, “CS is a broad discipline”. None of the teachers answered in such broad and unclear terms in the post survey. 11% of the teachers answered in the pre survey only that CS was related to ICT. Answers coded in this category included for example: “CS is related to ICT use”, “CS is the science that study technology”. Again, no teachers provided such answers in the post survey. Teachers viewing CS as a discipline that solve problems through technology, automation of tasks, and informatics systems, was an emerging answer exclusively in the post survey (38%). The wording “solves problems” was present in all of the answers coded in this category. For example some teachers mentioned: “CS is the discipline that helps solving all kinds of complex problems with computers in the areas of health, education, and security”.

During our teachers meetings we emphasized repeatedly that CS was not ICT and contrasted programming with ICT cognitive demands evidencing the logical thinking of programming vs the mechanical use of ICT. We believe that such emphasis resulted in teachers dissociating CS with ICT. Our approach to teaching programming through inquiry based programming challenges might have contributed to teachers perception of CS as a problem solving discipline.

Teachers also increased the reported number of CS topics they were familiar with between the pre and post survey. In the pre survey, half the teachers reported having no previous knowledge on any CS topics or having office automation skills (ex: I am a savvy internet and word processor user). In the post survey all teachers recalled PD content such as basic programming, basic algorithms, animation techniques with Alice, etc. They also mentioned specific concepts such as variables, conditionals, loops, parameter, attributes, etc.; all concepts learned along the PD course.

Survey responses indicate that teachers are gaining some understanding of CS subject matter, specially among teachers who had little or no previous knowledge on computer programming. Besides teachers learning of the subject, we also wanted to know how teachers brought what they learned into their classroom.

5.2 Bringing programming into the classroom

Teachers reported using more programming platforms in their classroom in the post survey than in the pre survey. Table 3 shows responses to the multiple choice question “What computer platforms do you use during your classes?” Teachers could choose more than one option.

Options	Pre-test	Post-test
My students do not use the computers	26%	16%
Word, spreadsheet, slide processors	71%	64%
Audiovisual production software	51.5%	48%
Spreadsheets or database programs	29%	40%
Programming languages platforms	27%	64%
Internet search	65%	78%

Table 3: Programs used in the classrooms (N=54)

We also asked teachers what percentage of what they usually learn in PD courses they bring into the classroom. Two times more teachers reported bringing between 90% and 100% of what they learned in "this" PD course into the classroom, suggesting that the characteristics of this PD course had effect in their teaching.

Besides teachers' self reported data on the survey, we want to describe different situations that contributed to our qualitative analysis. Every year about 300 students from participating schools come to our end of the year fair to show their animations and video games. Students' work is an important evidence showing that teachers are bringing programming into their classroom.

Teachers analysis of school CS curriculum required in our workshops plus supplying several CS teaching materials might have contributed to encourage teachers, and specially teachers with CS background who were not teaching programming, to change their practice. During our workshops teachers engaged in heated conversations when we asked what CS content knowledge schools should be teaching. Some teachers defended teaching to use commercial software, or preparing the workforce in one industrial programming language. Others advocated for teaching open source resources or teaching the basics of programming.

Because many teachers thought that teaching programming was not possible, this course represented a hinge moment in their professional career. One teacher expressed in the last survey:

"I started this course in a very particular moment regarding my profession: I am a System Analyst graduated in 1993, but started to "play" with computers in 1981-1982. I became proficient with Logo, Basic, Cobol, Pascal, Clipper, dBase, Fox. I was "disappointed" and a little frustrated, tired and "stuck" in my knowledge. This course made me fall in love again with my profession, I discovered that you can always start over and that that old knowledge can be dusted and rescued to support new content. I have much to thank the people of the National University of Cordoba because they literally changed my line of work and professional life."

Many other teachers have also told us similar stories. A great amount of teachers holding a CS background are not teaching CS because of curriculum mandates for ICT integration or for being unaware of pedagogical strategies for teaching CS in an engaging way.

5.3 Teachers implementation of the lessons

Coaches observations analysis showed differences in the way teachers brought CS into their classrooms. While most teachers selected the educational platform Alice to program animations or video games, 30% of the teachers taught CS concepts through robot programming and the rest used Chatbot or unplugged activities. Sometimes, teachers used more than one resource.

48% of the teachers used the same inquiry based programming challenges we used during our workshops. Teachers reproduction of the teaching practices they experienced as students is consistent with previous teacher learning theories. In this case, because a big part of the PD activities was devoted to solving challenges, could explain why a large percentage of teachers took these challenges to their classrooms However, 15% of the teachers presented students with open challenges that did little to develop understanding of CS concepts. For example, some teachers simply asked their students to make an animation using Alice, or a dialogue using Chatbot. In these lessons, the teachers did not design the tasks to focus on specific CS concepts. Lack of subject matter domain could explain why these teachers posed such broad challenges.

We did showed teachers other teaching platforms such as a Code.org and told them about the step-by-step lesson plans offered by them. However, it is clear that these strategies we "told" teachers did not compete with the strategies teachers experienced first hand. Only 30% of the teachers used strategies we showed but not experienced in the workshops, and 20% used unplugged activities we practiced during PD hours.

Two high school teachers at different schools decided to assemble and program their own robots using Arduino boards creating a "Programming Club" at their school. In this case teachers' previous technical knowledge combined with the new initiative resulted in a school based innovation.

5.4 Teachers explanations of CS concepts

Analyzing classroom observations we identified that in 75% of the lessons teachers tried to incorporate and explain fundamental CS concepts learned in the workshops. Analyzing each of the observed lessons we identified different levels of expertise explaining the concepts. Table 4 summarizes these levels in the 73 classroom observations.

Teacher could not explain the concept	5%
Teacher explained the concept by memory	3%
Teacher explained the concept correctly but commits a simple mistake	17%
Teacher explained the concept correctly	35%
Teacher explained the concept using analogies, examples and practical situations	23%
Other	13%

Table 4: Teachers explainations of CS concepts

58% of teachers explained CS concepts effectively. As an example of teachers explanation we cite one of the coaches reflections

"This is the first time that I see a teacher explaining the concept of methods and parameters. And he did it perfectly. Based on tutorial videos, but without using the video in class, he explained students how to make characters walk in Alice. He used method "pose" to capture all necessary movements. First, he explained how to create a method and asked how they thought they should do for a character to walk towards an object..."

The most frequent CS concepts that teachers were conditionals, loops, variable, sequence, methods, random numbers, objects and events. Teachers taught at least three of these concepts in about 70% of the lessons we observed. Parameters, constant, binary numbers, networks and other topics were less frequent and presented in about 30% of the lessons we observed. The inquiry based approach allowed to integrate more than one concept in a lesson. Almost 20% of the teachers confused CS concepts with platform commands. For example one of the teachers who decided to work with the Alice platform, confused the command of moving gallery objects, and choosing a scene as CS concepts. Out of 10 teachers who could not teach CS concepts correctly, 7 of them had no previous CS background, but 3 of them did,

including one primary school teacher with a certification in Educational Informatics. We are positive that this certification prepares teachers to use educational software in schools but do not address CS concepts. This finding suggest that teachers with no previous background on CS need longer teacher preparation than a 50 hours course.

6. CONCLUSIONS

Conducting a PD course for two consecutive years we have learned important lessons about teacher CS learning and classroom implementation of CS programming. Most teachers who come to our course have wrong or weak ideas about what CS is. For the last twenty years educational authorities and curriculum experts have conveyed the message that teaching CS is teaching office automation and ICT. Some teachers have CS background, but because of this curricular orientation they strongly believe do not have to teach CS. In many cases, because teachers had difficult CS learning experiences themselves and are unaware of newly developed teaching resources, they simply cannot imagine how to teach programming to primary and high school students. Understanding teachers beliefs and needs is necessary to design effective PD programs. Learning about the discipline CS and the importance of teaching it to promote both cognitive development and digital literacy, is our first challenge as trainers. Reflection and debates on teachers' practices contributed to change teachers pre notions about teaching CS. Teachers also learned about pedagogical content knowledge with first hand experiences on inquiry based programming teaching. Learning the pedagogy is as important as learning CS concepts because simply teaching algorithms and definitions can not promote meaningful learning of the discipline. Solving programming challenges during PD hours, and having a coach at their schools for their teaching practice, provided teachers with pedagogical resources to bring programming into their classroom. Requiring teacher classroom practices as part of the PD hours, was essential to promote classroom implementation. However, this training was insufficient to prepare teachers with no previous background on CS as they weakly addressed CS concepts in their classrooms. While we observed that teachers with previous CS background were better at explaining CS concepts in their classroom, it is unclear to us what specific CS background would be best to become an effective CS teacher. One strong limitation of this study is that we only observed lessons where the teachers were required to teach programming as part of the PD hours. In addition, all observed lessons were conducted in the presence of coaches who supported teachers and could have influence teachers' practices. Thus, we do not know about long term effects of our course beyond PD requirements. Nevertheless, the evidence presented here shows that a CS PD course that includes first hand programming experiences, curricular debates among teachers, and classroom practice with coaches support, contributed to increase teacher learning of CS and implementation of inquiry based programming lessons in schools.

Acknowledgments

This work was partially funded by the grants PICT-2014-1833, PICT-2012-712, PDTS-CIN-CONICET-2015-172, and PID-2012-2013-R18.

7. REFERENCES

[1] L. Benotti, M. J. Gomez, and C. Martinez. UNC++Duino: Learning to program in Python and C++ starting from blocks. In *Proceedings of the Conference on Robotics in Education*, 2016.

[2] L. Benotti, M. C. Martínez, and F. Schapachnik. Engaging high school students using chatbots. In *Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education*, pages 63–68, New York, NY, USA, 2014. ACM.

[3] J. Cuny. Finding 10,000 teachers. *CSTA Voice*, 5(6):1–2, 2010.

[4] B. Ericson, M. Guzdial, and M. Biggers. Improving secondary CS education: progress and problems. In *SIGCSE Bulletin*, volume 39, pages 298–301, 2007.

[5] P. A. Ertmer. Teacher pedagogical beliefs: The final frontier in our quest for technology integration? *Educational technology research and development*, 53(4):25–39, 2005.

[6] M. S. Garet, A. C. Porter, L. Desimone, B. F. Birman, and K. S. Yoon. What makes PD effective? Results from a national sample of teachers. *American educational research journal*, 38(4):915–945, 2001.

[7] J. Goode. If you build teachers, will students come? the role of teachers in broadening computer science learning for urban youth. *Journal of Educational Computing Research*, 36(1):65–88, 2007.

[8] K. Lang, R. Galanos, J. Goode, D. Seehorn, F. Trees, P. Phillips, and C. Stephenson. Bugs in the system: CS teacher certification in the US. *The CSTA and The ACM*, 2013.

[9] M. C. Martinez and M. E. Echeveste. Representaciones de estudiantes de primaria y secundaria sobre las ciencias de la computación y su oficio. *Revista de Educación a Distancia*, (46), 2015.

[10] P. Mishra and M. Koehler. Technological pedagogical content knowledge: A framework for teacher knowledge. *The Teachers College Record*, 108(6):1017–1054, 2006.

[11] A. Mühling, P. Hubwieser, and T. Brinda. Exploring teachers' attitudes towards object oriented modelling and programming in secondary schools. In *Proc of the Sixth international workshop on Computing education research*, pages 59–68. ACM, 2010.

[12] L. Ni and M. Guzdial. Who am I?: understanding high school CS teachers' professional identity. In *Proc of the Technical Symposium on Computer Science Education*, pages 499–504. ACM, 2012.

[13] N. Ragonis, O. Hazzan, and J. Gal-Ezer. A survey of CS teacher preparation programs in Israel tells us: CS deserves a designated high school teacher preparation! In *Proceedings of the 41st Symposium on Computer Science Education*, pages 401–405. ACM, 2010.

[14] D. Thompson and T. Bell. Adoption of new CS high school standards by New Zealand teachers. In *Proc of the Workshop in Primary and Secondary Computing Education*, pages 87–90. ACM, 2013.

[15] M. Windschitl and K. Sahl. Tracing teachers' use of technology in a laptop computer school: The interplay of teacher beliefs, social dynamics, and institutional culture. *American educational research journal*, 39(1):165–205, 2002.