"I Now Feel that this is Unfair" A Case Study on the Effects of Professional Development for Debugging in the K-12 Classroom

Tilman Michaeli[1(✉)] and Ralf Romeike[2]

[1] TUM School of Social Sciences and Technology, Computing Education Research Group, Technical University of Munich, Munich, Germany
tilman.michaeli@tum.de

[2] Computing Education Research Group, Freie Universität Berlin, Berlin, Germany
ralf.romeike@fu-berlin.de

Abstract. Finding and fixing errors is an essential skill in learning programming in the K-12 classroom. However, most of the time, debugging only plays a minor role in teachers' approaches to conveying programming - especially as they themselves rarely learned debugging explicitly and lack appropriate concepts and content. In consequence, students often struggle with finding and fixing errors on their own. Professional development allows for disseminating research findings and corresponding teaching materials to eventually influence the teaching practice. In this paper, we present a professional development workshop and its theoretical foundations, aiming at fostering teachers' professional competence with regards to debugging. We investigate changes in teaching practice and the teachers' beliefs in reaction to the PD using a case study approach. The results provide insights into impact and effects of professional development with regards to debugging in the classroom. Furthermore, our study contributes indications for designing professional development that fosters actual change in the classroom.

Keywords: Debugging · Professional development · Computing education · K-12 · Case study · Teaching practice

1 Introduction

Debugging can be considered a core problem in the K-12 classroom: novice programmers make more programming errors and, compared to experts, spend similar high amounts of time debugging [2]. Fixing errors is a significant obstacle to learning programming [20]. Helplessness and, in consequence, frustration when confronted with errors is a common phenomenon in the K12 classroom [26]. Accordingly, this is also a major challenge for teachers: They often rush from student to student, helping and trying to do justice to all of them as much as

A. Bollin and G. Futschek (Eds.): ISSEP 2022, LNCS 13488, pp. 78–89, 2022.
https://doi.org/10.1007/978-3-031-15851-3_7

possible [24]. Moreover, teachers - just like professional software developers [27] - have often not learned debugging systematically themselves and only seldom explicitly teach debugging [24]. In school practice, learners are therefore often left alone with their errors and are consequently forced to acquire appropriate strategies and approaches on their own. Experience has shown that this is a challenge that is hardly manageable for a large part of students [6]. However, explicitly teaching debugging has the potential to foster students' debugging self-reliance [3,7,25].

As to why they don't convey debugging skills in their classroom, teachers report that there is a lack of time - both in the classroom and for preparing appropriate concepts and materials. At the same time, teachers claim a lack of existing concepts, best practices, or materials for debugging in the classroom. Furthermore, they report that debugging is not an explicit part of the curriculum and therefore often neglected in favor of content explicitly required [24].

A traditional way to transfer educational innovations and achieve a change in the teaching practice is professional development (PD) for in-service teachers. However, designing effective PD provides a particular challenge, as typically in limited time, teachers' content and pedagogical content knowledge as well as beliefs need to be addressed - while demonstrating direct applications and implications for their pedagogical practice [22]. Therefore, in this study, we designed a corresponding professional development workshop and investigate the effects on the participants' teaching practice in the form of a case study to gain insights on how to tackle the core problem of debugging in the classroom.

2 Theoretical Background

The aim of PD is to improve the quality of teaching and thus the students' learning processes [9]. Research shows that this is highly dependent on teachers' *professional competence* which is considered an acquirable disposition [19]. Therefore, expanding teachers' professional competence is central for PD. It is comprised of cognitive, motivational and personal components [4]. Cognitive components are typically distinguished into content knowledge (CK), pedagogical knowledge (PK) and pedagogical content knowledge (PCK) based upon [31]. Concerning motivational factors of professional competence, teachers' beliefs in particular are considered crucial.

In order to achieve a change in professional competence, according to constructivism, the teachers' existing experience has to be taken into account [32]. Debugging describes the process of finding and fixing errors and is dependent on the underlying error type [15]. Debugging skills differ from general programming skills [1] and include the application of a systematic process, different debugging strategies, heuristics and tools [21,24,30]. Teachers themselves typically have rarely learned debugging explicitly and therefore lack appropriate concepts and content for teaching in class. Furthermore, they differ in their personal debugging process: While some teachers apply predominantly (basic) strategies such as *print* debugging and have little experience with tools such as the *debugger*, others even teach such "advanced" methods to their students [24]. With regard to

PCK, teachers hardly use explicit units to teach debugging skills in their lessons. Predominantly, they try to give the students assistance in individual support. Despite acknowledging the problem coping with errors provides for students, teachers lack the means and/or beliefs to eventually address them further in their classroom [24].

There are some studies investigating explicitly teaching debugging in secondary or tertiary [3,5,7,8,17,25], but to our knowledge there is no research done on PD specifically focusing on debugging in the classroom. However, there are extensive findings regarding PD in [14,23,28,29] and especially beyond [9,11,19,22] computing education research. In the following, we discuss a selection of core design principles from literature that seem especially important for our context.

Research shows that for an actual change in school practice, it is crucial to *create a common problem awareness*. To this end, it is important to work together with teachers on the basis of their teaching practice, instead of merely providing "top-down" materials. It is essential that the teachers' own teaching experiences are taken into account. For the implementation of concepts of the workshop in teaching practice, it is helpful to provide materials that are as *concrete as possible*, making them more or less directly-usable in the classroom. This enables teachers to use the materials directly without much effort and thus reduces the entry barrier. However, teachers typically adapt these concepts and materials to their specific needs in a second iteration at the latest. Therefore, teachers should be actively encouraged to experiment and adapt the materials as a starting point for teaching. Further success factors for PD are an interplay between theoretical input and practical explorations phases that allow for *self-directed and active learning*. Due to the heterogeneity of teachers' knowledge, they should be enabled to try out new debugging strategies in a self-determined way, for example, according to their individual learning requirements. Additionally, *collaboration, the exchange of ideas and networking* with and between the participating teachers is essential for a sustainable transfer of the workshop content into teaching practice.

Those principles derived from research on PD provide hints for designing general PD. However, designing a successful workshop is highly dependent on the content and context [22]. Therefore, research in PD on debugging and its effects on the teaching practice is necessary.

3 A PD Workshop on Debugging

Based on the principles established in research on general PD as well as findings on teachers' professional competence with regards to debugging, we designed and conducted a weekend PD workshop (Friday afternoon to Sunday noon) in November 2019 with 16 participating CS high school teachers from various regions of Germany. The event took place in a conference hotel, so that the participants stayed overnight. With the workshop, we aimed at fostering teachers' professional competence regarding debugging (in particular concerning beliefs,

CK and PCK), for eventually influencing their teaching practice. For the classroom materials used in the workshop, we mainly build upon a teaching concept for including debugging in the classroom, which effectiveness we investigated previously [25]. Within this concept, different types of errors are distinguished and a systematic approach for fixing them is proposed. Furthermore, concrete methods for the classroom are suggested, such as adapted debugging tasks, ways to introduce various debugging strategies and tools (such as print-debugging or the debugger), or an error glossary to help with forming patterns and heuristics (similar to professionals' debugging logs [27]).

Before the workshop began, teachers were asked to reflect on their personal debugging process and how they had learned to debug in the form of "debugging biographies" (see e.g. [18]). The evaluation of these biographies revealed that the majority of teachers never learned debugging systematically, but had mostly acquired the relevant skills on their own. Furthermore, it showed a large amount of heterogeneity with respect to the scope of the debugging strategies and tools used and also concerning the general programming experience outside the classroom.

Day 1: Awareness and self-reflection. At the beginning of the workshop, debugging was characterized as a core problem of teaching practice. To this end, the teachers first reflected how "typical" students proceed with debugging in their lessons and what problems they have using the *persona method* from Design Thinking [10] in groups (see Fig. 1). In doing so, they switched into the perspective of the learners to reflect on typical problems (a central component of *PCK*). In a second step, the groups exchanged the personas among themselves and described how they would support the respective student in their current teaching. This way, teachers should both share existing best practices and reflect on where they see a need to expand their teaching.

Day 2: Introduction and exploration of CK and PCK on debugging. At first, a professional software developer reported from his professional experience on how debugging works in industry, what significance it has and how developers "learn" debugging. In the next phase, the debugging process was formally introduced and corresponding debugging skills were systematized (*CK*). Furthermore, approaches to address respective required skills in the classroom were demonstrated (*PCK*). Afterward, the teachers worked through materials for classroom use. In doing so, they acquired corresponding *CK*, for example by working on materials for the introduction of debugging strategies with which they had little or no experience (such as the debugger). Furthermore, they directly tried out methods for teaching relevant skills and thus acquired according *PCK*. All ideas, suggestions, questions, and comments that arose during this phase were collected and then discussed in plenary.

Day 3: Transfer into their classroom. On the last day of the workshop, teachers were given the opportunity to transfer the concepts to their own teaching materials. Based on the personas, each teacher created a concrete plan of which ideas

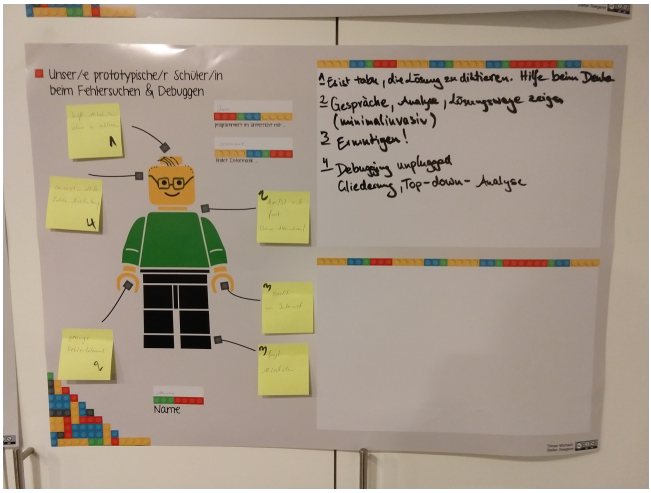

Fig. 1. Persona from the workshop

they wanted to take from the workshop and use in the classroom. These plans were then presented and discussed in groups.

Concluding reflection in the workshop. The design of the workshop was generally viewed positively by the teachers. Among other things, they highlighted the high proportion of active learning, the expert's input and the intensive networking among the teachers. The opportunity to try out concrete materials for the classroom was also positively evaluated. A clear consensus for the increased integration of the topic of debugging into teaching became apparent.

4 Methodology

This study aims to investigate the influence of the PD workshop on the actual teaching practice. Therefore, we address the following research question. **RQ:** How does the teaching of the participants involved in the PD workshop change with regards to debugging in the classroom?

There are many ways to evaluate the success of a PD workshop. Since we are explicitly interested in the transfer to teaching practice, we aimed to investigate the actual change in the teachers' classrooms – as opposed to an examination of the change in self-efficacy expectations or professional competence directly at the end of the training [22]. However, shortly after the workshop in November 2019, school practice was largely restricted for the survey period due to the Covid-19 pandemic. Given the severe additional challenges and demands for teachers – not just limited to remote teaching periods –, an in-detail comparative analysis does not appear to be expedient. Furthermore, it is to be expected that the time available and willingness to implement new concepts was severely limited.

Therefore, under these special conditions, a qualitative case study methodology was chosen to investigate the research question. This allows for a precise analysis of cases for teachers reporting to have adapted their teaching in consequence of the workshop, despite or even before the pandemic effects on schools. Furthermore, we can examine the characteristics of the individual cases and the respective circumstance in detail to comprehensively map the changes in the teaching practice [33].

Data Collection and Case Selection. Towards the end of the school year, semi-structured interviews were conducted online with the participants of the workshop. In the process, *critical cases* [33] were selected, which allows to check the connection between the workshop and changes in the teaching practices. Despite the Covid-19 pandemic, two of the teachers contacted had already integrated corresponding content from the PD workshop into their lessons to a significant extent before the school closures began, or had implemented it in remote teaching despite the corresponding challenges. Both were well-experienced CS high school teachers from different German states (with different curricula and ways of anchoring CS as a subject). Within the semi-structured interviews, the teachers were asked whether and how their teaching changed in consequence to the PD, in particular with regards to their awareness of and reaction to students' problems, as well as whether they tried out the methods proposed – and if so, which experiences they made.

Data Analysis. The interviews were first transcribed and evaluated in a case-by-case analysis (*within-case analysis*) to develop a deeper understanding of the respective changes [12]. Subsequently, central *cross-case* characteristics were identified.

5 Results

In the following section, the two cases are described in detail, in order to then identify common characteristics afterwards. All quotes have been translated into English by the authors with minimal adjustments to improve comprehensibility.

5.1 Teacher I

Teacher I reports that shortly after the end of the workshop, he already implemented the first concepts regarding a systematic debugging approach with year ten students:

> *As soon as that was possible, I tried it. [...] We categorized [errors] by type and also discussed how to cope with them. There were actually two lessons where that was the topic. The students had a given programming project, and there were errors built-in, and they had to solve different tasks, [...] from a semicolon that was missing to semantic errors that appeared at the end, even though the program was running perfectly.*

He applied debugging tasks and let the students categorize different error types and emphasized that a different approach was necessary for different kinds of errors. In general, categorizing errors with the students represents one of the central changes in the teaching. The teacher was also able to give his assessment of success and impact on the students before the Covid-19-based distance learning:

How helpful this is in the long run is difficult to confirm or disprove at the moment. For the weaker students it has definitely been helpful. They have at least gratefully accepted this categorization. And they [...] then wrote an error glossary, and I think they pulled it out again when they had errors that they didn't know about at first and had to remember.

With the joint error collection in the form of a "glossary", another idea of the workshop was adapted and successfully implemented. The teacher's experience indicates that "weaker" students in particular have actually benefited from the support. In general, the teacher emphasizes that in the future he wants to introduce the handling of errors at the beginning of the programming lessons.

In addition, the teacher has transferred debugging to the topic of spreadsheets and also used "debugging tasks" here:

So it was interesting to see what kind of errors there were. They often have problems there as well. You have to run around because too many hash marks are displayed, which simply indicates that the columns are too narrow. Typical errors are division by zero and so on. And the error messages are quite cryptic.

Dealing with errors is an overarching theme for the teacher after the workshop and is not limited to programming. This emphasizes the general educational importance of debugging beyond programming. As a consequence of the workshop, the teacher focuses especially on the error culture in class:

So another aspect we covered in this context in the lesson before Christmas, but that I have also done occasionally in recent years, is to discuss famous software bugs. Making errors, Ariane and Mars Lunar Voyager and so on, all that stuff. These are exciting stories, which also show the students that even on a large scale, mistakes are made and that errors just happen.

After reflecting his teaching, he assumes that there is a connection between the opening of tasks and the perception and handling of errors:

The tasks are often such that the students have to solve a problem very specifically. The task is clearly defined. The students have to solve it and then stumble somewhere into these mistakes they then make and are then disappointed because it doesn't work out right away. [...] This means that in every lesson you get to the point when the students make mistakes: "Now I have made a mistake. I didn't solve the problem the way the teacher

intended." And this should actually be different in the whole programming class: that I work more creatively and simply give the students more freedom in the exercises.

In the future, he therefore wants to use more open assignments in the initial lessons to test what influence this has on how students deal with their errors. This can be described as an *productive-failure* approach [16]. Overall, the importance of the topic for teaching has thus increased and is also multiplied in his training of teachers:

From practice I can still say that I am now also addressing this issue with teacher trainees. With them, I will cover this as a explicit topic in the next few weeks, because currently we were hindered by Covid. But now, until the end of the school year, it is also my goal that we explore approaches [for debugging in class].

In addition, the teacher also reports that his own approach to debugging has evolved in the context of programming projects in a newly-learned programming language:

There I learned something I never did and never needed in Java, which we discussed at the workshop, namely print debugging. I need it once or twice for some Java problems, but with the Python problems it was so massive. [...] That means, you always have to look, what is input, what is output, what is the current state of the variables? And I did an incredible amount of print debugging. Which I have never actually done before.

In summary, the teacher reports a significant change in his classroom as a result of the PD workshop. Coping with errors is now perceived as a content that spans all topics, students are supported particularly well by systematizing and collecting various error types, and the teacher tries to create a positive error culture in the classroom.

5.2 Teacher II

The second teacher also reports that he has integrated debugging more prominently in his teaching as a result of the workshop. Accordingly, he introduced two debugging strategies for the first time during distance learning. First, a "debug class", which logs the calls of the methods of a given project:

I introduced a debug class, because I am looking for errors myself and with the help of this debug class, traces were created and these traces should be displayed in sequence diagrams to use the aspect of modeling [...], but also to make basic mechanisms clear. That means we did not learn from the error, but from the trace.

Thus, this debugging strategy was initially not introduced in the context of error correction, but rather in the context of learning about modeling, and only then should it be used for debugging. In addition, the debugger was introduced as an optional task in analogy to the approach of the teaching concept presented in the workshop, using a project created by the teacher. The teacher emphasized that these two approaches came from the workshop, since debugging is not an explicit topic in the curriculum.

He hopes that this change will give the students more autonomy in debugging, which will also enable them to carry out other kinds of teaching projects – for which he was unable to gain experience because of distance learning:

> *This is the innovation in my teaching or in my approach. That was not the focus in the past. The point was to keep the project so small that I actually assume that it would be flawless.*

In conclusion, the teacher summarizes the changes in his teaching practice, especially concerning debugging:

> *What you have done, from my point of view, is you have influenced the beliefs of our colleagues. In consequence, I see the importance better than before now. I actually think that it is not quite fair to always reproach the student for not finding the errors when I have not even shown him what I normally use. I now feel that this is unfair. That's why the minimum I have to show him is what I usually use myself. I usually use tracing, so that's what I teach.*

This shows a changed view on his teaching, which emphasizes that without the teaching of adequate strategies, students are not able to deal with errors on their own. The workshop has thus contributed to changing the teacher's beliefs concerning the importance of debugging for programming lessons.

In summary, the importance of debugging in class – although not anchored in the curriculum – has increased significantly for this teacher. Based on his own debugging approach, he now systematically introduces debugging strategies to increase the students' independence.

6 Discussion

Comparing the two cases concerning the change in teaching, the first thing to be noted is the increased **value** of debugging. The importance of debugging in the classroom – even beyond programming – is reflected in (increased) time spent on explicitly conveying debugging skills by the teachers.

Neither teacher has directly adopted the materials from the workshop, but both have **adapted** them for their own needs (in line with literature [13]). In doing so, they have set different priorities: For example, teacher I focused particularly on the aspect of error culture and the categorization of different types of errors, while teacher II primarily conveyed his personal debugging approach and

strategies to students. In both cases, this results in a **extension** of the workshop contents: For example, teacher I transfers the concepts of the PD to the teaching of spreadsheets, while teacher II combined the introduction of appropriate tracing strategies with teaching modeling.

As a consequence of the workshop, both teachers also **reflect** their teaching practice with regard to dealing with errors: They would like to try out more open tasks in the future. Teacher II hopes improving students debugging skills allows him to use such open formats. Teacher I suspects a connection between more open tasks and the perception and handling of errors according to a *productive-failure* approach, which he would like to explore in the future.

The basis for these changes in teaching practice is the **expansion of professional competence** in consequence of the workshop: On the one hand, the teachers have acquired corresponding *content knowledge*, which even influenced their personal debugging process. On the other hand, the teachers learned about different ways to teach debugging skills (*pedagogical content knowledge*). At the same time, the *beliefs* of the teachers have changed towards the explicit teaching of debugging. In both cases, this had hardly played a role in their teaching practice before.

Limitations. Due to the Covid-19-related teaching situation and the corresponding challenges in the classroom, many participants of the workshop reported that they had not yet integrated debugging content into their lessons as planned. A broader evaluation of the PD regarding the transfer into the teaching practice was not possible. However, the cases of teachers who had already implemented debugging in their classes before or despite Covid19-circumstances allow for deep insights into the changes in their teaching and thus implications about success factors of the workshop.

7 Conclusion

In this paper, we investigated the effects a PD workshop on debugging in the K-12 classroom has on the teaching practice of attending teachers. For this purpose, a three-day workshop was designed based on general research findings regarding PD, as well as research on teachers' professional competence with regards to debugging in the classroom. Given the pandemic situation, we conducted a case study to analyze effects on participants' teaching. In consequence, our results provide deep insights into two teachers' change in practice.

The analysis revealed that the workshop had the intended effect for the two cases. This was particularly evident in the increased importance of the topic in teaching practice. In most cases, the materials of the workshop were the starting point for individual adaptation according to personal needs and different foci. In addition, the awareness on the significance of errors for the learning process introduced in the workshop sparked further ideas for extensions, such as transferring and combining the concepts with other topics.

Furthermore, the results provide indications for designing professional development that foster actual change in the classroom. To this end, our results

suggest that convincing the teachers of the importance of the topic (teaching debugging in the classroom) is crucial: even if not explicitly required in the curricula, debugging is an essential part of the programming process. Fostering students' debugging skills even offers teachers the potential to improve their own teaching practice further, such as by including more open exercises. Furthermore, the teachers' reports indicate that it is precisely the reflection of their own teaching practice that has contributed to the creation of an awareness of the problem and was thus essential for changing teachers' beliefs towards debugging. Actual change in practice was supported by the concrete materials as possible starting points for teaching debugging skills: The teachers adapted the ideas of the workshop to their personal needs, expanded them and experimented with them – even beyond the programming classes.

References

1. Ahmadzadeh, M., Elliman, D., Higgins, C.: An analysis of patterns of debugging among novice computer science students. In: Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, pp. 84–88. ACM, NY (2005)
2. Allwood, C.M., Björhag, C.G.: Novices' debugging when programming in Pascal. Int. J. Man Mach. Stud. **33**(6), 707–724 (1990)
3. Allwood, C.M., Björhag, C.G.: Training of Pascal novices' error handling ability. Acta Physiol. **78**(1–3), 137–150 (1991)
4. Blömeke, S., Felbrich, A., Müller, C., Kaiser, G., Lehmann, R.: Effectiveness of teacher education. ZDM **40**(5), 719–734 (2008)
5. Böttcher, A., Thurner, V., Schlierkamp, K., Zehetmeier, D.: Debugging students' debugging process. In: 2016 IEEE Frontiers in Education Conference (FIE), pp. 1–7. IEEE, Erie, PA (2016)
6. Carver, S., Klahr, D.: Assessing children's logo debugging skills with a formal model. J. Educ. Comput. Res. **2**(4), 487–525 (1986)
7. Carver, S., Risinger, S.: Improving children's debugging skills. In: Empirical Studies of Programmers: Second Workshop, pp. 147–171. Ablex Publishing Corp. (1987)
8. Chmiel, R., Loui, M.C.: Debugging: from novice to expert. ACM SIGCSE Bull. **36**(1), 17–21 (2004)
9. Clarke, D., Hollingsworth, H.: Elaborating a model of teacher professional growth. Teach. Teach. Educ. **18**(8), 947–967 (2002)
10. Dahiya, A., Kumar, J.: How empathizing with persona helps in design thinking: an experimental study with novice designers. In: IADIS International Conference Interfaces and Human Computer Interaction (2018)
11. Darling-Hammond, L., Hyler, M., Gardner, M.: Effective Teacher Professional Development. Learning Policy Institute, Palo Alto, CA (2017)
12. Eisenhardt, K.M.: Building theories from case study research. Acad. Manage. Rev. **14**(4), 532–550 (1989)
13. Farmer, J., Gerretson, H., Lassak, M.: What teachers take from professional development: cases and implications. J. Math. Teacher Educ. **6**(4), 331–360 (2003)
14. Goode, J., Margolis, J., Chapman, G.: Curriculum is not enough: the educational theory and research foundation of the exploring computer science professional development model. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education, pp. 493–498 (2014)

15. ISO, I.: IEEE, systems and software engineering-vocabulary. IEEE Computer Society, Piscataway, NJ 8, 9 (2010)
16. Kapur, M.: Productive failure. Cogn. Instr. **26**(3), 379–424 (2008)
17. Katz, I., Anderson, J.: Debugging: an analysis of bug-location strategies. Hum. Comput. Interact. **3**(4), 351–399 (1987)
18. Knobelsdorf, M., Schulte, C.: Computer biographies-a biographical research perspective on computer usage and attitudes toward informatics. In: Proceedings of the Koli Calling 2005, pp. 139–142. ACM, NY (2005)
19. Kunter, M., Kleickmann, T., Klusmann, U., Richter, D.: The development of teachers' professional competence. In: Kunter, M., Baumert, J., Blum, W., Klusmann, U., Krauss, S., Neubrand, M. (eds) Cognitive Activation in the Mathematics Classroom and Professional Competence of Teachers. Mathematics Teacher Education, vol. 8, pp. 63–77. Springer, Boston (2013). https://doi.org/10.1007/978-1-4614-5149-5_4
20. Lahtinen, E., Ala-Mutka, K., Järvinen, H.M.: A study of the difficulties of novice programmers. ACM SIGCSE Bull. **37**(3), 14–18 (2005)
21. Li, C., Chan, E., Denny, P., Luxton-Reilly, A., Tempero, E.: Towards a framework for teaching debugging. In: Proceedings of the Twenty-First Australasian Computing Education Conference, pp. 79–86. ACM, NY (2019)
22. Lipowsky, F., Rzejak, D.: Key features of effective professional development programmes for teachers. Ricercazione **7**(2), 27–51 (2015)
23. Menekse, M.: Computer science teacher professional development in the united states: a review of studies published between 2004 and 2014. Comput. Sci. Educ. **25**(4), 325–350 (2015)
24. Michaeli, T., Romeike, R.: Current status and perspectives of debugging in the K12 classroom: a qualitative study. In: 2019 IEEE Global Engineering Education Conference (EDUCON), pp. 1030–1038. IEEE, Dubai (2019)
25. Michaeli, T., Romeike, R.: Improving debugging skills in the classroom: the effects of teaching a systematic debugging process. In: Proceedings of the 14th Workshop in Primary and Secondary Computing Education, pp. 1–7. ACM, NY (2019)
26. Perkins, D.N., Hancock, C., Hobbs, R., Martin, F., Simmons, R.: Conditions of learning in novice programmers. J. Educ. Comput. Res. **2**(1), 37–55 (1986)
27. Perscheid, M., Siegmund, B., Taeumel, M., Hirschfeld, R.: Studying the advancement in debugging practice of professional software developers. Softw. Qual. J. **25**(1), 83–110 (2016). https://doi.org/10.1007/s11219-015-9294-2
28. Qian, Y., Hambrusch, S., Yadav, A., Gretter, S.: Who needs what: recommendations for designing effective online professional development for computer science teachers. J. Res. Technol. Educ. **50**(2), 164–181 (2018)
29. Ravitz, J., Stephenson, C., Parker, K., Blazevski, J.: Early lessons from evaluation of computer science teacher professional development in Google's CS4HS program. ACM Trans. Comput. Educ. **17**(4), 1–16 (2017)
30. Rich, K.M., Strickland, C., Binkowski, T.A., Franklin, D.: A K-8 debugging learning trajectory derived from research literature. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, pp. 745–751. ACM, NY (2019)
31. Shulman, L.: Knowledge and teaching: foundations of the new reform. Harv. Educ. Rev. **57**(1), 1–23 (1987)
32. van Dijk, E.M., Kattmann, U.: A research model for the study of science teachers' PCK and improving teacher education. Teach. Teach. Educ. **23**(6), 885–897 (2007)
33. Yin, R.: Case Study Research: Design and Methods, 3rd edn. Sage, London (2003)