Analyzing teachers' diagnostic and intervention processes in debugging using video vignettes

Heike Wachter[0000−0002−5090−7183] and Tilman Michaeli[0000−0002−5453−8581]

Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany
{heike.wachter,tilman.michaeli}@tum.de

Abstract. Providing individualized support to students during debugging is a huge challenge for teachers in K-12 computing education. In these everyday assessment situations, they often have little time to gather relevant information to diagnose the student's problem and respond with an appropriate intervention. Thus, diagnostic and intervention processes in debugging are essential for teachers. Despite the importance, there is a lack of research on this topic and its possible implications for the classroom. Therefore, this paper aims to provide insights into teachers' diagnostic and intervention processes in debugging. In this qualitative study, we investigate situation-specific aspects teachers consider for diagnosing error situations and interventions they apply in a specific debugging-related situation using video vignettes. To this end, scripted video vignettes depicting a typical classroom debugging situation were presented to experienced teachers, who reported their observations in open-ended questionnaires. The data were then analyzed using qualitative content analysis. The results show a wide range of different aspects used in diagnostic processes and in proposed interventions. Furthermore, our results indicate that teachers rarely address motivational and emotional aspects of debugging in their interventions. These findings contribute to a better understanding of teachers' diagnostic and intervention processes and how they can be fostered in teacher education.

Keywords: debugging · K-12 · computing education · diagnostic and intervention process · video vignettes

1 Introduction

Debugging – that is, finding and fixing errors in program code – is an important and unavoidable part of programming [29]. However, for novice programmers, finding and fixing errors is a huge challenge [3] and a barrier to learning programming. In K-12 computing education, students often rely on the teacher for support [3]. In consequence, teachers tend to rush from student to student, trying to get them to solve the problem on their own in a short period of time [25]. Thus, for teachers in K-12 classrooms, individual support for students in debugging is a major challenge as well: On the one hand, the teachers want to support the students so they don't feel neglected. Still, on the other hand, the teachers also want the students to try to debug their program themselves rather than having the solution dictated to them or even typed out for them.

While supporting a student, the teachers go through a multi-step diagnostic and intervention process (cf. [18,17]). First, the teacher must identify the problem and why the student is stuck. Based on this diagnosis, the teacher has to consider a suitable intervention. The assistance should be minimal to foster the student's self-reliance [17]. Finally, the teacher carries out the intervention and evaluates whether the student was able to solve the problem successfully.

Diagnostic and intervention skills are considered as a core element of teachers' PCK and professional competence [1]. However, pre-service teachers often experience difficulties practicing diagnostic and intervention processes in their first years of teaching [5]. Therefore, the development of diagnostic and intervention skills plays a central role in successful teaching and should be fostered in teacher education [4].

However, there is little scientific evidence on teachers' diagnostic and intervention processes in debugging. Detailed information about the processes and how they can be fostered is necessary to promote diagnostic and intervention processes in teacher education. Therefore, the present work aims to gain insights into the processes using video vignettes. Thus, we investigate situation-specific aspects that teachers consider in their diagnosis and their proposed intervention to understand teachers' diagnostic and intervention processes and how they can be fostered.

2 Background and Related Work

Diagnostic process. In the daily teaching practice, teachers are often confronted with spontaneous assessment situations in which they need to gather information about their students' learning prerequisites, processes, and outcomes [5]. The diagnostic process is defined as the assessment of different situations in which knowledge is applied to solve problems and make decisions [11]. Typically, diagnostic processes focus on identifying aspects or characteristic expressions of individuals [1]. Thus, diagnostic processes in teaching mean the assessment of relevant aspects for student learning [2]. Therefore, teachers need to recognize students' different learning needs, abilities, interests, and motivations [36].

Diagnostic skills are an essential element of teachers' professionalism [1] and also referred to as a key competence [18], as a successful diagnostic process enables teachers to provide individual support to their students and to adapt their teaching to students' needs [36].

In debugging situations, teachers must recognize and assess relevant situation-specific aspects to identify the problem and make a diagnosis. Doing so, they build on knowledge about typical errors, misconceptions, and other problems in introductory programming [30]. They have to understand why the student made an error and cannot solve it independently. A lack of content knowledge, interest, or knowledge of debugging strategies [25,13] could be the reason. Further, motivational and emotional aspects can be crucial in students' debugging processes [16]. Based on this diagnosis, teachers can anticipate various possibilities of support, taking the student's ability and level of knowledge into account [15].

Intervention process. The results of diagnostic processes can be used to determine subsequent measures, so-called *interventions* [33]. An intervention is a minimal interference in the student's problem-solving process, which enables the student to overcome a barrier in the learning process and continue working independently [20]. In addition, suitable interventions do not focus only on problem-solving but consider the cognitive, motivational, and emotional aspects of learning. [33]. With the help of appropriate interventions, high student achievement can be attained [37]. Therefore, teachers need an extensive repository of interventions [17] and the skill to choose the right intervention for each situation.

In debugging, the teacher implements an intervention based on the previous diagnosis to help the student overcome the difficulties in finding and fixing the error. When deciding on a suitable intervention, the teacher has to consider the student's learning level [15] and how much support is necessary, as the assistance should be minimal to foster the student's self-reliance [17]. Further, there is evidence of features for suitable interventions [17]. In line with current research on intervention features (cf. [33]), debugging interventions must be effective and efficient. The aim is to ensure that students not only grasp the necessary concepts to work independently but also receive timely assistance with minimal time and effort, allowing teachers to attend to the needs of all students equitably.

Investigating diagnostic and intervention processes in debugging. Teachers' diagnostic and intervention skills are rarely investigated in computing education. Looking at debugging, existing research focuses on the skills needed for debugging and how these can be taught to students in the classroom. Various studies investigate the debugging behavior of novices, providing evidence on novices' typical errors or difficulties [27], helpful methods and approaches that can support learners in debugging [26], and skills and knowledge needed for effective debugging [25].

However, in contrast to automated approaches to diagnostics and feedback in programming [7], there is little evidence on how teachers diagnose and intervene in class. Tsan et al. [35] investigated teachers' PCK in debugging after a training with Scratch. They concluded that teachers often supported their students with code-level solutions when encountering incorrect code. Furthermore, some preliminary evidence exists on teachers' interventions when supporting students in the K-12 classroom [13]. Concerning student-teacher interaction for debugging, Nixon et al. [28] analyzed two teachers' approaches to support students facing uncertainties in engaging with physical computing systems. They observed strategies the teachers applied in these situations, such as asking questions, articulating the problem, or giving directions. Similarly, Hennessy Elliott et al. [12] analyzed the pedagogical approach of a middle school teacher supporting her students in learning to debug physical computing systems. In particular, they identified pedagogical possibilities used by the teacher for supporting students, such as emphasizing a systematic process that prepares students for potential problems, affective responses to students' frustration and joy, or positioning themselves as learners alongside their students during programming and debugging. In summary, there is only very limited evidence of teachers' diagnostic and intervention processes (and their connection) in debugging.

Video vignettes. Video vignettes are a common and effective approach for evidence-based teacher education [9,32] and investigating diagnostic and intervention processes [5]. As specific teaching-learning behaviors are often difficult to observe in natural classroom settings [32], researchers have developed scripted video vignettes [5] to facilitate observation of these aspects. Such video vignettes have the advantage that they offer *representational scaffolding* in contrast to the complex situation in the classroom [8]. Furthermore, they can be viewed multiple times and with different emphases, which helps foster pre-service teachers' PCK [10]. Video vignettes are particularly used in other educational sciences such as maths [34], physics [19], or biology.

3 Methodology

This work investigates situation-specific aspects teachers consider and assess when diagnosing students' debugging problems and the intervention they would suggest in this situation. Therefore, we address the following research questions:

- **RQ1:** *Which situation-specific aspects do teachers consider when diagnosing students' problems?*
- **RQ2:** *Which interventions do teachers propose based on their diagnosis?*

Video vignettes. We employed a video vignette approach with questionnaires to answer the research questions. To this end, we presented teachers with scripted video vignettes showing typical teaching situations. This allows us to investigate teachers' diagnostic and intervention processes in a particular situation as video vignettes simulate typical classroom situations realistically [9]. In contrast to screenshots and program files where only information from the code is available, the teacher has all the situational aspects available for diagnostic purposes that they would have in an actual classroom situation. In addition, unlike field observations, this approach makes the data collected more comparable, as all participants responded to the same situations.

We used four scripted video vignettes showing typical debugging situations for a pair of students in the classroom. The particular situations are derived from literature on common novice problems and behaviors and from practical experience (see [38]). All vignettes have a duration between 3 and 4 minutes. The behavior, communication, and content of the screen (see figure 1) of two students working together and struggling with a particular error are presented. The content of the vignettes is described below.

In *video vignette 1* (*V1*), two students program a ball in the Java development environment Greenfoot. However, their program contains a syntax error and thus can not be executed. The students identify the incorrect class and try to fix the error independently. But they insert the missing bracket in the wrong place. As a result, the semicolon is missing as a closing argument, and Greenfoot returns an

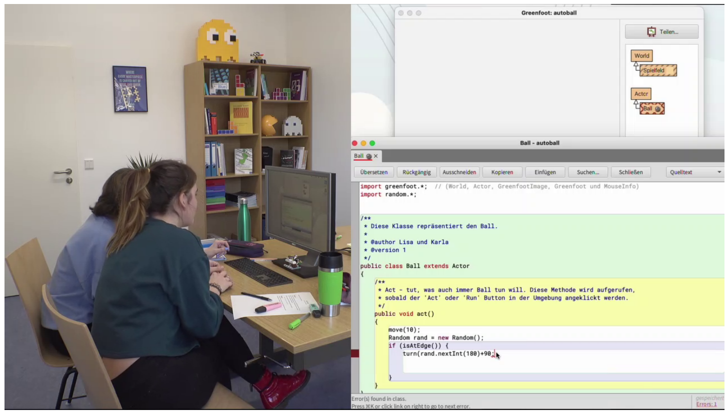

Fig. 1. Screenshot of Vignette 1

error. In *video vignette 2* (*V2*), the students get multiple error messages caused by an erroneous library import. The incorrectly integrated library leads to an error and two subsequent errors when calling the library functions. They switch between the errors and read the error messages but do not understand them. *Video vignette 3* (*V3*) shows how the students create an object ball and receive a null pointer exception. They do not understand the error message and try to close the window several times. Finally, in *video vignette 4* (*V4*), the students try to implement a boost function for the ball and receive another null pointer exception. The students are unsure about how to proceed since the object exists.

We asked teachers participating in the study to rate the authenticity of the teaching situations shown in the video vignettes. Based on the statement, "The video vignette shows an authentic teaching situation," teachers used a 5-point Likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree) for rating.

For all video vignettes, we received confirmation of predominantly realistic classroom situations (mean: 3,86, median: 4).

Data Collection and Analysis. Data was collected in the context of three professional development workshops on debugging with K-12 computer science teachers in Germany. We collected the data at the beginning of the workshops to prevent data bias and to ensure validity. For each vignette, the student's behavior was presented twice, with a five-minute gap in between to take notes. Afterward, the participants had time to complete the questionnaire.

A total of 23 teachers participated in the study. We did not pre-select them, so we had one teacher with little experience in Java programming. However, most of the teachers had advanced knowledge of Java programming and at least five years of teaching experience in grades 7 to 12. Given time constraints, not every vignette was used in all three workshops. In summary, we received 15 completed questionnaires for V1 and V3, 14 for V2, and 13 for V4.

In this workshop setting, we considered questionnaires with an open response format most appropriate for data collection, addressing the limited time available but still enabling open response options. The questionnaire was structured as follows: In the first question, the teachers were asked to describe which aspects of the students' behavior they noticed in the video. The second question deals with why the students cannot solve the problem on their own. The teachers should give their diagnosis and briefly explain how they came to this conclusion. Finally, in question 3, the teachers were asked to explain how they would react as teachers in the respective teaching situation. The questions 1 and 2 refers to RQ 1 and question 3 to RQ2.

To analyze the data, we conducted a qualitative content analysis according to Mayring [23] with inductive category formation. The analysis was software-supported with MAXQDA. Before data analysis, we cleaned the data set of empty and meaningless responses. Then, a category system was developed inductively for the remaining answers. Therefore, each new code was checked to determine whether it could be classified in an existing category or, if not, a new category was created. If, on the other hand, a new coding fits into more than one category, the concerned categories were too specific. They were then combined into a new category with an appropriate name. The first author mainly carried out the coding. For reliability testing, a second researcher independently coded 10% of the data. The categories created intuitively in this way largely corresponded to the first author's categories. If there was a discrepancy, both researchers decided which category name best reflected the content. The intercoder reliability showed a match of 70% (Cohen's Kappa coefficient: 0.70) [24].

4 Results

RQ1: Which situation-specific aspects do teachers consider when diagnosing students' problems?

First, we analyzed the aspects the teachers noted for diagnosing students' problems. Thus, we developed a category system with aspects considered in the teachers' diagnostic process. These categories are described below.

Content knowledge: Commonly, teachers identified a lack of content knowledge and its application in a particular situation as reasons for the students' struggling. For example, teachers noted a "lack of syntactic knowledge", "lack of basic understanding", or "terms and meaning seem unclear".

Problem-solving strategies: In addition, teachers identified aspects concerning problem-solving strategies as one of the reasons why students could not solve the problem independently. Many teachers described the students' approach as "trial and error using hints from the program" or "changing the code without much reflection (trial and error)". Especially in the context of dealing with error messages, the teachers noted the student's lack of any systematic approach to cope with them. One teacher described this as "they lack the experience to 'interpret' the error messages".

Emotion: An emotion describes the expression of feelings, such as joy or anger. Some teachers mentioned students' emotional reactions when dealing with debugging problems. For example, in vignette 4, the teachers recognized the students' joy at their progress ("joy of students about own success"). They also mentioned their frustration when the following error occurred ("error is recognized and generates strong emotional reaction (frustration)").

Motivation: Motivation is the process of setting and pursuing goals and is often influenced by emotions. Teachers noted motivational aspects regarding the students' debugging in the video vignettes. They mentioned that students were initially keen to fix the error but quickly gave up. One teacher described the initial motivation as a "visible desire to find a solution". More teachers mentioned the students' resignation, for example, "giving up quickly" or "the students call the teacher immediately to get help". Some teachers concluded that this lack of motivation was why the students could not solve the problem. For example, they mentioned "because they do not try" or "only half-hearted participating (one student)" as reasons.

Comparing the teachers diagnosis, we found a large variance in the teachers' responses. Teachers perceived different situation-specific aspects when viewing a vignette. Thus, different diagnoses were made for the same debugging situation. Many teachers focused on aspects of content knowledge and debugging strategies during the monitoring. In contrast, motivational and emotional aspects were mentioned only occasionally.

RQ2: Which interventions do teachers propose based on their diagnosis?

Analyzing the teachers' answers on the interventions they would apply in the specific debugging situation, we categorized the interventions as follows:

Asking questions: One kind of intervention teachers noted, in our data, was to use questions to lead students. We can distinguish between two types of questions: questions at the beginning of the interaction with the student that might serve to quickly gather information about the current problem while still leading the students, for example, "Where does what error occur?" or "What do you want the code to do?". The other type of questions our data reveals are several consecutive questions. We suppose these series of questions were used to guide the student in finding and fixing the error. Therefore, the teachers proposed a series of questions that build on each other to guide the students toward solving the problem. An example of this was for Vignette 2: "What are libraries? Why do we need them? How can we use libraries? What are the consequences of not loading libraries? Were there really three errors?". This type of intervention was mentioned by many teachers regardless of the situation presented in the vignette.

Giving hints: Another frequently mentioned intervention is giving hints to students on how to solve problems. Teachers use hints, for example, to help students isolate an error: "give hints, where the wrong parenthesis belong to" or "suggest what the problem could be (typical errors)". To solve the problem, hints on programming concepts or keywords in the code are suggested ("remember the keyword new!").

Explain: Another type of intervention we identified is to explain concepts or procedures. Teachers reported that they would explain necessary concepts such as imports ("explain what imports are for") or clarify procedures ("explain to students that an object 'ball' must first be created") to support students. Sometimes, these are illustrated with examples ("make students aware of the correct syntax by giving an example").

Problem-solving together: Furthermore, many teachers proposed a collaborative approach to supporting students. For example, they noted down "executing the code step by step to understand the logic" together with the students so that they could understand where the error occurred. Some teachers also suggested teaching students a systematic approach to debugging by doing it together, such as "let the students mark structural units in the code → does each structure have everything required?".

Encourage independence: Some teachers considered it essential to encourage students' independence. One teacher, e.g., noted: "encourage them to find their own solutions". Furthermore, teachers proposed to "encourage students to use some of the existing solutions" or strategies they already know. Another approach in this category was to "let them analyze the error message" and "look up commands individually" or to "let students refer to syntax (look it up)".

Discuss in class: Some teachers suggested discussing errors that occur for the first time or affect a large number of students in the class: "discussion in the class when an error occurs for the first time".

For all vignettes, we found that the teachers often did not suggest only one type of intervention in a particular debugging situation but combined different approaches. For example, for the syntax problem in vignette 1, one teacher noted down to first give a hint about where the missing parenthesis should go and then explain why the compiler does not show the correct location for the parenthesis. A closer look at the chosen interventions reveals that teachers primarily focus on aspects regarding content knowledge and problem-solving strategies. Regarding content knowledge teachers suggested: "referring students to syntax", "looking up commands one by one", "explaining import procedure", and "explaining the structure of a control condition". In terms of problem-solving strategies, teachers suggested, for example, "discussing a systematic approach", "letting the students explain the procedure and encouraging them to use some of the solutions already available", and "making suggestions about what could be the reason (typical errors)" or drawing attention to program features ("pointing out the red wave"). At the same time, motivational and emotional aspects that impact self-directed learning are rarely considered within the interventions. For example, in video vignette 3, the students try to solve the error independently and persist for a long time. Most teachers did not note this aspect as part of their diagnosis, and no teacher mentioned praising them for their effort in their intervention. Considering the intervention chosen in the context of the situation presented and the diagnosis made, we find that only some teachers chose interventions tailored to the situation and diagnosis. In one case, for example, the teacher noted in his diagnosis, "The students show a lack of prior knowledge. They cannot categorize the terms." and his suggested intervention was to "run the program step by step to understand the logic". Furthermore, many teachers chose rather general intervention strategies that work in different situations, e.g., "discuss systematic approach" or "analyze error message together", regardless of their diagnosis.

5 Discussion

To gain insights into teachers' diagnostic and intervention processes in debugging, we investigated them through video vignettes. The teachers rated the authenticity of the video vignettes with an average acceptance value of 3.86 and median of 4 on a 5-level Likert scale (1: "strongly disagree", 5: "strongly agree"). Thus, we assume the video vignettes reflect the teachers' practical experience and authentically resemble typical classroom situations.

Our results show a large variance in the considered situation-specific aspects and the resulting diagnosis among the teachers. Thus, teachers made different diagnoses for the same video vignette. We were surprised by these results, and we can't exactly explain where the differences came from. Possible reasons for the differences could be different backgrounds, different experiences, or different qualifications of the teachers. Which should be investigated in future work.

Furthermore, our data indicates that teachers often focused on aspects of content knowledge and problem-solving strategies but rarely noted motivational and emotional aspects. Content knowledge and problem-solving strategies address central components of programming and debugging skills [31] and thus are essential to consider. However, there is evidence that motivational and emotional aspects are particularly important for supporting students in debugging. Debugging can be seen as a productive failure [14] and thus involves an interplay between emotion, motivation, and learning [22]. Various studies investigated students' emotional reactions toward debugging (e.g., [16,6]). They found that emotional reactions, especially negative ones, impact students' performance, and self-efficacy [21]. Thus, we consider motivation and emotion essential aspects for teaching to consider while diagnosing and intervening. Hennessy Elliott et al., e.g., present a teaching approach in debugging, considering these aspects [12].

The intensity of support in teachers' proposed interventions also varied considerably between teachers. Some focused on promoting students' independence and suggested only minimal assistance. Others proposed actively accompanying the students' solution process by discussing the code. These results are in line with [25]. The differences in the suggested intervention may originate from the video vignette situation and the respective representational scaffolding. In the actual teaching practice, interventions are also heavily dependent on the available time for one particular student, as one teacher noted: "Depending on the time available: general advice if there is much time, more specific advice the less time is available."

One perspective to analyze teachers' proposed intervention can be whether they focused primarily on fostering students' process or the product [12]. While the former aims to support the student's debugging process in the longer term with its intervention, the latter is primarily concerned with helping the student solve the current problem and achieve an executable program. In our data, we found both dimensions. For example, one teacher suggested "reading and understanding the code with the students (taking it apart)". This collaborative approach helps the student to understand the error and how to deal with it – so the focus is on the process. Another teacher suggested giving the students a hint that "the semicolon and the bracket are arranged incorrectly". Here, the focus of the proposed intervention is on solving the problem quickly and, therefore, on the product. Overall, we saw a focus on the product for most suggested interventions in our data.

Limitations. A possible limitation of this study is the difference between the video vignettes and the actual classroom situations. Participants have much more time to focus on individual cases and see a much longer extract of the students working on their problem (about 3-4 minutes) as possible in the teaching practice. In addition, video vignettes miss the teacher-student relationship. Usually, teachers know their students and their characteristics. These may influence the diagnosis as well as their proposed interventions. However, the representational scaffolding provided by this approach allows detailed insights into the participants' processes. Moreover, teachers rated the situations presented in the video vignettes as authentic.

A methodological limitation could be that we used questionnaires with an open-ended response format to conduct the study. Interviews could have provided a deeper insight into the diagnostic and intervention process due to their interactive nature and the possibility of asking follow-up questions. Therefore, building on these results, we plan to extend the study with interviews.

6 Conclusion and Future Work

This paper provides findings on teachers' diagnostic and intervention processes in debugging. To this end, we presented scripted video vignettes to teachers showing specific situations with students struggling with debugging. This allowed us to analyze how teachers diagnose particular problem situations and what interventions they suggest to support their students in these situations.

Our results show a large variance in the situation-specific aspects considered for the diagnosis, and different diagnoses for the same situation. Furthermore, our data indicates that teachers often base their diagnoses on aspects related to content knowledge and problem-solving strategies, while motivational and emotional factors are only rarely considered.

The interventions suggested by the teachers to deal with the problem situation provide insights into the teachers' intervention repertoires and the selection of an intervention for a particular diagnosis. As with the diagnosis, our data shows that teachers rarely considered motivational and emotional aspects in their interventions. Furthermore, the data indicates that many teachers chose interventions that work in different situations; only some teachers chose interventions tailored to the situation and diagnosis.

Despite their importance in teachers' daily practice, diagnostic and intervention skills (not only for debugging) have received very limited attention within computing education. We believe that it is essential to foster such skills in teacher education and professional development. To this end, our findings contribute to our understanding of teachers' diagnostic and intervention processes in debugging and provide valuable implications. In particular, we consider it crucial to emphasize the importance of emotional and affective components in diagnosing and intervening in the classroom. Similarly, our findings highlight the importance of striking a balance between helping students get unstuck (product focus) and promoting their autonomy in debugging by providing appropriate strategies (process focus). In general, the apparent disparity and variety of diagnoses and interventions we found in our data points to the complexity of this process – which, in turn, underscores the need for supporting teachers.

In future work, we aim to investigate the quality of teachers' diagnostic and intervention processes. Furthermore, we want to evaluate the effectiveness of a video vignette approach for fostering diagnostic and intervention skills in debugging in teacher education.

References

1. Aufschnaiter, C.v., et al.: Diagnostic competence: Theoretical considerations concerning a central construct of teacher education. ZfPäd **61**(5), 738–758 (2015)
2. Binder, K., et al.: Diagnostic Skills of Mathematics Teachers in the COACTIV Study, pp. 33–53. Springer International Publishing, Cham (2018)
3. Carver, S.M., Klahr, D.: Assessing children's logo debugging skills with a formal model. JECR **2**(4), 487–525 (1986)
4. Chernikova, O., et al.: Simulation-based learning in higher education: a meta-analysis. Review of Educational Research **90**(4), 499–541 (2020)
5. Codreanu, E., et al.: Between authenticity and cognitive demand: Finding a balance in designing a video-based simulation in the context of mathematics teacher education. Teaching and Teacher Education **95**, 103146 (2020)
6. Dahn, M., DeLiema, D.: Dynamics of emotion, problem solving, and identity: Portraits of three girl coders. Computer Science Education **30**(3), 362–389 (2020)
7. Danielsiek, H., et al.: Undergraduate teaching assistants in computer science: Teaching-related beliefs, tasks, and competences. In: EDUCON. pp. 718–725 (2017)
8. Fischer, F., et al.: Representational scaffolding in digital simulations–learning professional practices in higher education. ILS **123**(11/12), 645–665 (2022)
9. Gamoran Sherin, M., Van Es, E.A.: Effects of video club participation on teachers' professional vision. Journal of teacher education **60**(1), 20–37 (2009)
10. Gaudin, C., Chaliès, S.: Video viewing in teacher education and professional development: A literature review. Educational research review **16**, 41–67 (2015)
11. Heitzmann, N., et al.: Facilitating diagnostic competences in simulations in higher education: A framework and a research agenda. FLR **7**(4), 1–24 (Oct 2019)
12. Hennessy Elliott, C., et al.: Toward a debugging pedagogy: helping students learn to get unstuck with physical computing systems. ILS **124**(1/2), 1–24 (2023)
13. Hennig, H., Michaeli, T.: Investigating teachers' diagnostic and intervention skills in debugging. In: Proceedings of the 17th WIPSCE. ACM (2022)
14. Kafai, Y.B., et al.: Rethinking debugging as productive failure for cs education. In: 50th ACM TS SIGCSE. pp. 169–170 (2019)

15. Karst, K., Dotzel, S., Dickhäuser, O.: Comparing global judgments and specific judgments of teachers about students' knowledge: Is the whole the sum of its parts? Teaching and teacher education **76**, 194–203 (2018)
16. Kinnunen, P., Simon, B.: Experiencing programming assignments in cs1: the emotional toll. In: Proceedings of the Sixth ICER. pp. 77–86 (2010)
17. Klock, H., Siller, H.S.: Measuring an aspect of adaptive intervention competence in mathematical modelling processes. In: CERME11 (2019)
18. Klug, et al.: Diagnostic competence of teachers: A process model that accounts for diagnosing learning behavior tested by means of a case scenario. TATE **30** (2013)
19. Krumphals, I., et al.: Fostering pre-service physics teachers' diagnostic skills and readiness through video vignettes and micro-teaching sessions: An exploratory single-case study. Book of Extended Abstracts p. 364 (2023)
20. Leiss, D.: Adaptive teacher interventions in mathematical modelling—empirical findings of a comparative laboratory and classroom study. JMD **31**, 197–226 (2010)
21. Lishinski, A., et al.: Students' emotional reactions to programming projects in introduction to programming: Measurement approach and influence on learning outcomes. In: ACM ICER. pp. 30–38 (2017)
22. Litts, B.K., et al.: Perceptions of productive failure in design projects: High school students' challenges in making electronic textiles. In: Singapore: ISLS. ISLS (2016)
23. Mayring, P.: Qualitative content analysis: theoretical foundation, basic procedures and software solution. SSOAR, Klagenfurt (2014)
24. McHugh: Interrater reliability: the kappa statistic. Biochemia medica **22** (2012)
25. Michaeli, T., et al.: Current status and perspectives of debugging in the k12 classroom: A qualitative study. In: EDUCON. pp. 1030–1038. IEEE (2019)
26. Michaeli, T., et al.: Improving debugging skills in the classroom: The effects of teaching a systematic debugging process. In: WIPSCE. pp. 1–7 (2019)
27. Michaeli, T., et al.: Investigating students' preexisting debugging traits: A real world escape room study. In: Koli Calling '20. ACM (2020)
28. Nixon, J., et al.: Teachers' learning to support students during science inquiry: Managing student uncertainty in a debugging context. In: ICLS. pp. 601–608 (2023)
29. Perscheid, M., et al.: Studying the advancement in debugging practice of professional software developers. Software Quality Journal **25**(1), 83–110 (2017)
30. Qian, Y., et al.: Students' misconceptions and other difficulties in introductory programming: A literature review. TOCE **18**(1), 1–24 (2017)
31. Robins, A., Rountree, J., Rountree, N.: Learning and teaching programming: A review and discussion. Computer science education **13**(2), 137–172 (2003)
32. Seidel, T., et al.: Developing scripted video cases for teacher education: Creating evidence-based practice representations using mock ups. In: Frontiers in Edu (2022)
33. Severing, E., Weiß, R.: Quality development in research on vocational education, vol. 12. W. Bertelsmann Verlag (2013)
34. Sommerhoff, D., et al.: Pre-service teachers' learning of diagnostic skills in a video-based simulation: Effects of conceptual vs. interconnecting prompts on judgment accuracy and the diagnostic process. Learning and Instruction **83** (2023)
35. Tsan, J., et al.: An analysis of middle grade teachers' debugging pedagogical content knowledge. In: ITiCSE. pp. 533–539. ACM (2022)
36. Vogt, F., Rogalla, M.: Developing adaptive teaching competency through coaching. Teaching and teacher education **25**(8), 1051–1060 (2009)
37. Weinert, F.E., Schrader, F.W., Helmke, A.: Quality of instruction and achievement outcomes. International Journal of Educational Research **13**(8), 895–914 (1989)
38. Zoppke, T., Michaeli, T., Romeike, R.: Individuelle unterstützung beim debuggen–video-vignetten für die lehrkräftebildung. INFOS 2023 (2023)