Towards Learnersourcing Relatable and Contextualized Learning Materials: An Exploratory Study in a Database Programming Class

Tianyi Li
Purdue University
West Lafayette, United States
li4251@purdue.edu

Yuzhe Zhou
Purdue University
West Lafayette, United States
zhou922@purdue.edu

Alejandra J. Magana
Purdue University
West Lafayette, United States
admagana@purdue.edu

Abstract

Engaging students from diverse backgrounds in computing education necessitates relatable learning materials. While instructors often possess deep knowledge in computing subjects, understanding the unique cultural and experiential backgrounds of students can be both time-consuming and challenging. This paper investigates the use of learnersourcing to overcome these barriers. We adopted constructivist principles to create a learnersourcing workflow that enables students to (1) connect their personal background with computing topics, (2) craft contextualized problems, (3) develop step-by-step solutions, and (4) engage in self and peer feedback. We evaluated this workflow in a database programming class and gathered student-generated worked examples, along with evaluations from peers, instructors, and students themselves using four criteria: difficulty, clarity, relevance, and educational value. We found that the students were able to identify new contexts and apply newly acquired computing knowledge in a meaningful way. However, the results also highlighted the need for better scaffolding to improve the clarity of problem description, ensure the availability and sufficiency of example data, and check for errors in the step-by-step solutions. Furthermore, student self- and peer assessments aligned with instructor evaluations of student-generated content. Reflecting on lessons learned, we outline six design implications for future improvements of the learnersourcing workflow.

CCS Concepts

• **Social and professional topics** → **Computing education**; • **Human-centered computing** → *Empirical studies in collaborative and social computing*.

Keywords

student-centered learning, contextualization, learnersourcing, constructivism, collective knowledge creation

ACM Reference Format:
Tianyi Li, Yuzhe Zhou, and Alejandra J. Magana. 2025. Towards Learnersourcing Relatable and Contextualized Learning Materials: An Exploratory Study in a Database Programming Class. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2025), June 27-July 2, 2025, Nijmegen, Netherlands.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3724363.3729117

ITiCSE 2025, June 27-July 2, 2025, Nijmegen, Netherlands

ACM ISBN 979-8-4007-1567-9/2025/06
https://doi.org/10.1145/3724363.3729117

1 Introduction

When teaching computing concepts, an important component is to demonstrate how these concepts are applied to solve real-world problems [3]. For example, database programming classes usually have a running example where students would download example data and learn how to use various SQL queries to derive insights from the data. However, one example scenario can hardly resonate with all students. Worse still, the examples used in computing courses can sometimes perpetuate historical and systematic biases [9, 23, 37] and thus may appear irrelevant and unrelatable to students of underrepresented backgrounds [9]. Developing learning materials that connect with diverse student backgrounds plays an essential role in broadening participation in computing [2, 27].

An important philosophy to address this challenge is student-centered learning (SCL), in which students take an active role in determining what, when, and how they learn [35]. The challenge is that effectively collecting and synthesizing student input is not trivial. Learnersourcing, which evolved from the concept of crowdsourcing, is a method in which learners collectively create new content for future learners, thus enriching their own learning experience in the process [17]. Previous research has demonstrated the effectiveness of learnersourcing in promoting student-centered learning by allowing them to generate various types of content, such as personalized hints for debugging support [12], typical misconceptions when learning new subjects [13], multiple choice questions (MCQ) [8] and other exercises [16]. Although learnersourcing has been effective in helping students identify common mistakes and create targeted exercises, it remains underexplored whether students can also reflect on and contribute to the contextualization of learning materials within diverse and relatable contexts.

In this paper, we investigate how to engage students in contextualizing computing concepts within scenarios that are relatable to their own experiences. We adopt the learnersourcing approach to design a workflow that breaks down the process of contextualizing learning materials into a series of learning activities, each with step-by-step guidance and instructions. The design of this learnersourcing workflow is guided by Gance's constructivist principles [11], viewing students as experts in their own backgrounds and interests, as well as experts in training in the computing subjects they are learning. We use worked-out examples (WEs) as the primary format for learning materials. A worked-out example is a group of step-by-step instructions on how to solve a problem usually generated by an expert in a specific area [31].

We evaluated the feasibility of a proposed learnersourcing workflow in a database programming class with 23 junior and senior students at a public land-grant university in the United States. Our investigation focused on two research questions:

RQ1: How well do students perform in the tasks of connecting their unique backgrounds to the computing subject, formulating contextualized problems, and developing step-by-step solutions?

RQ2: How well can student self-assessment and peer assessment evaluate the quality of student-generated worked-out examples?

We found that the students were able to identify new contexts and apply newly acquired computing knowledge in a meaningful way. However, the results also highlighted the need for better scaffolding to improve the clarity of problem description, ensure the availability and sufficiency of example data, and check for errors in step-by-step solutions. Furthermore, student self- and peer assessments aligned with instructor evaluations of student-generated content. Reflecting on lessons learned, we outline six design implications for future improvements of the learnersourcing workflow.

In summary, this paper contributes (1) an empirical understanding of the processes and challenges when students contextualize computing concepts in their own scenarios; (2) a learnersourcing workflow that guides students to contextualize computing concepts and create relatable programming learning materials; (3) design implications for future learnersourcing workflows and tools to support inclusive programming education.

2 Related Work

2.1 Contextualizing Learning Materials

Contextualization has been defined as "a diverse family of instructional strategies designed to seamlessly link the learning of foundational skills and academic or occupational content by focusing teaching and learning squarely on concrete applications in a specific context of interest to the student" [21].

Contextualizing computing learning materials in relatable scenarios has been shown to motivate novice programmers and facilitate skill transfer [4]. Additionally, it helps students understand the practical utility of computing knowledge, which is crucial for reducing dropout rates [14]. Students demonstrate greater autonomy, associated with intrinsic motivation in self-determination theory [7], when they perceive a meaningful connection between the learning materials and their personal interests [30]. This perception encourages students to invest more time and effort in learning the subject and pursuing the field [19, 26].

However, creating an applicable, appropriate, and relatable contextualization is not only resource-intensive, but also challenging, considering the varying backgrounds of instructors and the diversity of student populations [37, 39]. To address this challenge, we explore ways to engage students create contextualized learning materials related to their own experiences and interests.

2.2 Engaging Students in Co-creating Learning Materials with Learnersourcing

The act of creating resources and knowledge for peers underscores the social nature of learning, emphasizing collaboration and shared experiences [25]. This can also lead to the Protege Effect [5], where individuals gain benefits from teaching others, emphasizing that explaining or tutoring enhances the understanding and mastery of the material for the teacher [33].

Rooted in crowdsourcing and participatory learning [17], Learnersourcing is an innovative artifact to engage students in co-creating learning materials. It transforms students from passive consumers to active contributors and co-creators of knowledge [1, 36].

Prior research has applied learnersourcing in various programming classes and subjects. Guo et al. learnersourced student-created explanations of misconceptions while they are coding in a Python class [13]. The student insights were found to expand expert blind spots as programming instructors. Glassman et al. [12] introduced two workflows engaging learners in creating hints in a computer architecture class, fostering reflection on their work and peers' contributions. Guided by the supporting software, students were able to create hints that augment or even replace teachers' personalized assistance. Pirttinen et al. [29] investigated the use of a learnersourcing system in the context of teaching SQL. They found that the learnersourced content could cover the topics specified by the instructor and that students were able to engage with the learnersourced exercises.

Building on the prior successes, we further explore how to use learnersouricng as an artifact to guide students to contextualize and develop more complex learning materials.

3 Workflow Design and Conceptual Framework

We design a learnersourcing workflow guided by the constructivist principles. Constructivism posits that knowledge is actively constructed by the student rather than passively absorbed from textbooks and lectures [28]. Contextualization between knowledge and the learners' experiences can be considered as a constructivist process, as knowledge transfer is facilitated by participation in authentic tasks anchored in meaningful contexts [32].

Applying constructivism principles requires careful consideration of guidance and scaffolding for the "constructing" process [6]. Thus, each component of the workflow is designed in correspondence to a key pedagogical components within the constructivist principles proposed by Gance [11] (Figure 1).

The first component engages learners in learning new information and connects to the existing course environment, such as the regular lectures and lab exercises. The goal is to help student "warm up" and revisit the core concept learned in the class. This could take the form of multiple choice questions (MCQ), fill in the blanks, or other question formats relevant to the course topic.

The second component focuses on ideating scenarios and contexts where the computing concepts can be applied, and the scenarios and contexts should reflect students' own interests and experience. A constructivist pedagogy often demands the creation of authentic problem-solving situations. Traditionally, instructors bear the responsibility of conducting research and collaborating with local partners to contextualize learning materials [22, 24]. In this step, students will actively participate in the identification and framing of these contexts themselves.

The third component guides learners to apply the identified contexts by developing worked-out examples. In constructivist pedagogy, active engagement typically involves hands-on, dialogic interactions. The pivotal feature of this component lies in its "dialogic"

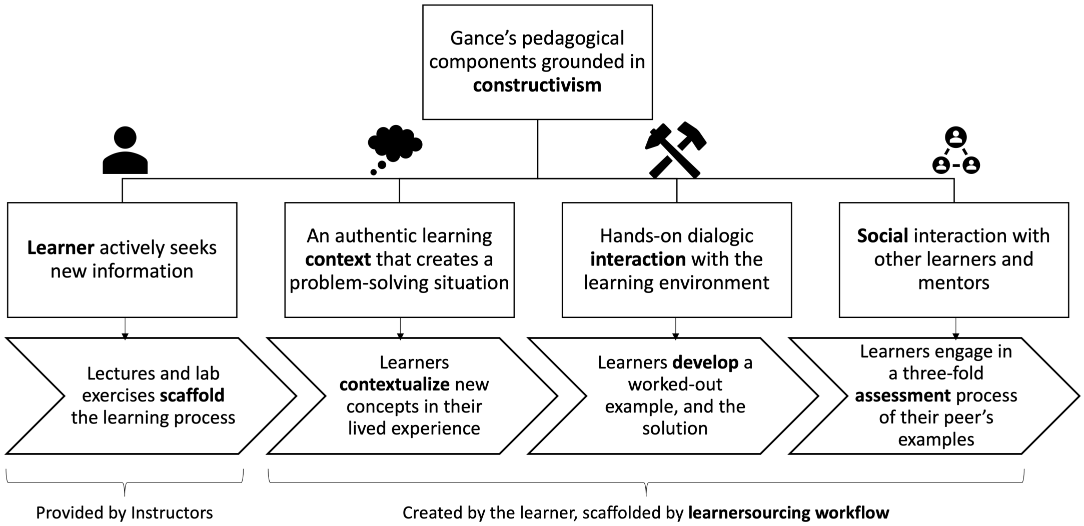

Figure 1: Mapping between the constructivist pedagogical components [11] and our learnersourcing workflow design.

nature, where learners face challenges and overcome obstacles in the process of constructing knowledge. We use the construction of work-out examples to implement this "dialogic" interaction, where students are required to (1) formulate a problem statement that pertains to specific course topics and (2) devise step-by-step solutions to the problem. Such sequentially dependent subtasks would prompt students to reflect on and refine their work in each step.

The fourth component is to learnersource assessments for the learnersourced materials. Constructivist learning environments encompass a social dimension that involves meaningful interaction among learners themselves and with mentors. The fourth component implements this social dimension through a two-fold evaluation process: self- and peer-assessment. Self-assessment encourages reflection, while double-blind peer assessment fosters translucent social interaction to support mutual awareness and accountability among students [15]. The peer assessment also allows students to interact with their classmates by working on and evaluating each other's worked-out examples.

Overall, the learnersourcing workflow is designed to engage students not only as learners of the computing concepts but as experts of their lived experience in applying course topics to their familiar environment. Through the peer assessment process (component 4), students have the opportunity to glean insights from each other's worked-out examples (components 2 and 3), thereby enriching their own understanding (component 1) and actively contributing to the collective knowledge of the entire class.

4 Research Testbed and Procedure

We experimented with the learnersourcing workflow in an advanced database programming course at a large public university. The course topic is PL/SQL, a Procedural Language extension for SQL (Structured Query Language) developed by Oracle.

Student Population and Course Schedule. Most students in this course are juniors or seniors majoring in Information Technology or related fields. The course has a prerequisite of an introductory SQL course. The class typically enrolls 20-25 students during Spring and 40-50 Fall, reflecting a 1:4 female-to-male ratio that is consistent with the college's overall demographics. The study was conducted in Spring 2022 with 23 students enrolled.

Course Organization and Content Each week, students attended two 50-minute lecture sessions and one 2-hour lab session over a 16-week semester. Lecture sessions focused primarily on introducing new concepts through in-class examples and interactive demonstrations. Lab sessions offered semi-guided exercises requiring students to apply their knowledge by solving problems contextualized within a database system designed for a simulated company specializing in electric parts. The course content was divided into three major modules: (1) PL/SQL basics: PL/SQL is a block-structured language, encompassing variables, SQL statements, loops, constants, conditional statements, and exception handling; (2) Functions and procedures: these are stored-program grouping SQL and PL/SQL statements for specific tasks, stored in a user's schema; (3) Triggers: a stored program automatically executed when some events occur.

4.1 Researchers Positionality and Limitations

The study protocol was reviewed and approved by the author's Institutional Review Board. Two authors are the instructor and the teaching assistant for this course. Another author is an engineering education researcher with over 15 years of experience in design-based research. A limitation of our approach was the collection of data in a regular course. Despite offering real-world insights, we were unable to implement a between-subject design due to fairness considerations. The instructor author also recognizes her potential influence on students' participation and responses.

To mitigate these limitations, the education researcher was deeply involved in the design of the learnersourcing workflow. This includes critically evaluating and aligning the workflow design with the constructivism conceptual framework, to ensure its theoretical validity and practicality. To avoid biases in the data analysis, worked-out examples (WEs) were graded based on effort rather than instructor ratings during the semester. In-depth data analysis on the de-identified WEs was conducted more than three months after the final grade was submitted. This allowed sufficient time for the instructor and teaching assistant authors to decouple from their instructor's role and approach the analysis more objectively.

4.2 Instruction Design and Data Collection

We implemented the first three components of the learnersourcing workflow as a bonus point opportunity within an online quiz on the topic of "functions and procedures" and the fourth component as a bonus point assignment.

The online quiz comprises multiple choices and fill-in-the-blank questions. These questions were designed to revisit and assess the students' understanding of PL/SQL functions and procedures. This serves as the first component of the workflow.

In the first part of the bonus question, students were asked to design an easy and medium question to practice applying the concepts of functions and/or procedures. They needed to outline a scenario for the question and provide the necessary tables, including the schema and actual records. The medium difficulty question should utilize solutions from the easy question. An instructor example was provided for both easy and medium difficulty questions. This two-step instruction was intended to scaffold students to formulate problems that are nontrivial and to facilitate the development of the step-by-step solutions. This part serves as the second component, as well as the first half of the third component of the workflow.

In the second part of the bonus question, the students were asked to provide their suggested solution for the easy question and then their solution to the medium question. This serves as the second half of the third component of the workflow.

At the end of the bonus question, the student self-assessed the questions they created. This serves as the self-assessment part of the fourth component of the workflow.

After the quiz, the instructor and teaching assistant compiled the students' problem descriptions and solutions into formatted worked-out examples (WEs). These WEs were then randomly assigned to students for peer evaluation in the form of a bonus assignment. The students were instructed to work on the WEs and then to evaluate them with the same rubrics used in the self-assessment. If the WEs has quality problems and cannot be used, the evaluation should also describe the issues and suggest possible improvements.

Finally, the teaching assistant assessed all WEs using the same rubrics. The ratings were reviewed and approved by the instructor.

4.3 Assessment Rubrics.

We crafted assessment rubrics to evaluate the WEs from four aspects: difficulty, clarity, relevance, and educational value.

Difficulty was assessed with three levels: "Not Difficult," defined as any one who can pass this course can answer, "A Bit Difficult," defined as people who can get C or higher can answer, and "Difficult," defined as only people who of A grade can answer. For clarity, the rubric ranged from "Not Clear," indicating potential confusion when another student reads it, to "Clear," denoting questions easily understood by the classmates. The relevance dimension assessed how well the questions pertained to the use of functions and procedures, distinguishing between questions that could optionally use these elements and those that required them for a solution. Finally, the educational value was gauged from "Not Helpful/Educational," where the questions contributed nothing to learning about PL/SQL, to "Very Helpful/Educational," which significantly enhanced PL/SQL skills. Each dimension included a "No Participation" option for those who did not submit a question.

4.4 Data Analysis

We evaluated the quality of student-generated worked-out examples (WEs) from both educator and learner perspectives. From the educator's viewpoint, the authors performed a thematic analysis on the problem statements and solutions in the WEs. This analysis aimed to assess the diversity and relevance of the contexts and the accuracy of the solutions provided. To capture the learners' perspectives, we analyzed the feedback provided during peer assessments, focusing on comments related to the usability of the WEs.

Categorize the Contexts Used in Worked-Out Examples. Two authors independently extracted keywords from the problem descriptions that best represented the context. Following this, each author conducted independent open coding and grouped similar keywords to form categories. Subsequently, the authors compared and synthesized the categories, ensuring each category was specific and different, covering all the keywords from the WEs.

Rate the Relevance of Worked-Out Examples. One author labeled the components in the solutions that represented the topic of functions and procedures. The labels were reviewed and approved by another author. The quality of the data and solutions was assessed by whether the code would compile and execute, the number of errors, and whether sufficient and meaningful data were provided.

Measure Rating Alignment. Given the ordinal nature of our evaluation measures and the relatively small sample size, we employed Friedman's Chi-Square Test [10] to examine the consistency of ratings across different groups. We also calculated the effect sizes [1] to quantify these differences. We also employed Krippendorff's alpha [2] to measure inter-rater agreement among peer assessors for the same worked-out examples (WEs).

5 Results

Twenty-three worked-out examples (WEs) were collected. Two WEs were excluded from the peer review process; one was a duplicate of the instructor's example, and the other presented identical one-sentence questions for both easy and medium difficulty levels. Of the remaining WEs, 16 received the five reviews, while five received only four reviews. In total, 100 peer reviews were collected.

5.1 RQ1: Feasibility and Support

5.1.1 Context Diversity. Our thematic analysis revealed five categories of contexts: video games (N = 1), sports (N = 1), university majors, classes and grades (N = 4), sales, customers, products, and orders (N = 11), employees and departments (N = 6). The last two themes resembled the examples provided by the instructor, which

[1] Effect sizes for the Friedman test are measured using Kendall's W coefficient, which follows Cohen's interpretation guidelines: 0.1 - < 0.3 (small effect), 0.3 - < 0.5 (moderate effect), and >= 0.5 (large effect)

[2] Krippendorff's alpha is interpreted according to guidelines suggested by Krippendorff: $\alpha \geq 0.800$ indicates good inter-rater reliability, $0.800 > \alpha \geq 0.667$ allows only for tentative conclusions, and $\alpha < 0.667$ suggests poor interrater reliability. [18]

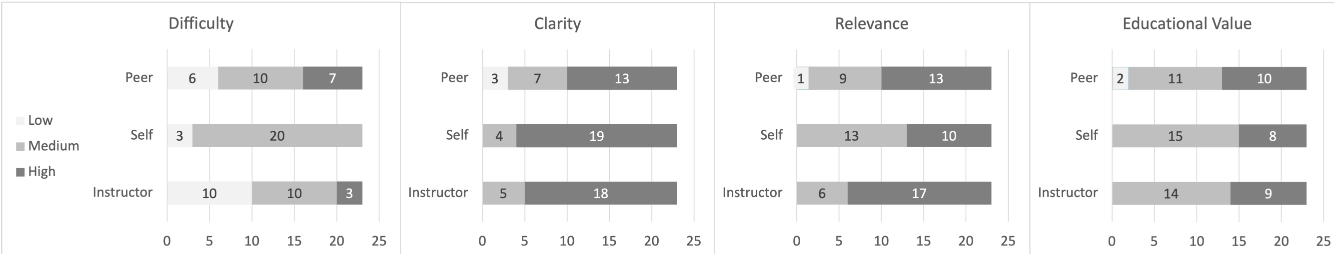

Figure 2: Self, Peer, and Instructor Ratings of WEs. Light color indicate lower levels, e.g. low difficulty. Zero values are omitted.

focuses on an electronic parts company with employee, inventory, and sales management data.

Although representing different contexts, some WEs lacked detailed background information (N = 15). For example, one WE's problem description was "write a function that, given the month, returns the calculated total revenue". While categorized as the theme "sales, customers, products, and orders" with the keyword "revenue", the scenario was relatively generic.

In contrast, three worked-out examples (WEs) were contextualized in specific and novel scenarios. One WE utilized PL/SQL to tackle a video game-related challenge, using PL/SQL to identify the Universal Product Code (UPC) and trace the origin of a game. Another example involved a soccer match between Liverpool and Manchester City, using PL/SQL to write a function that determines the maximum goals scored by players. A third WE focused on restaurant operations, using PL/SQL to identify servers who work in their home state.

5.1.2 Problem Relevance. All WEs included components that represent the topic of PL/SQL functions and procedures. The solutions required extracting or synthesizing specific information based on given inputs. For instance, one WE's problem description is "Write a procedure that, given a class name, identifies the highest GPA in that class." The associated solution used variable declaration, data types, value assignment, SELECT INTO statements, and aggregate functions like "MAX".

5.1.3 Data and Solution Quality. Around a third of the students (N = 8) provided descriptions and examples of the data needed to solve the problem they formulated. The other students either did not describe the table (N = 10) or did not provide example data (N = 5). Most students (N = 14) provided solutions that were executable, while eight solutions had syntax errors. One student described the answer but did not provide the code.

5.1.4 Usability of the Worked-Out Examples. The results of the peer assessment shed light on quality issues from the learners' perspective. Comments from peer assessments revealed three primary types of confusion experienced by students when engaging with WEs created by their peers. Firstly, some students found the problem descriptions to be overly vague, making it challenging to understand the task requirements. For instance, one student commented, "Wordiness was a slight mistake in the question since I was unsure of what actually to do." Secondly, a recurring issue was the insufficient data within the exercise, as one student remarked, "There is only one customer for each area; more data needed." Lastly, confusion arose when the exercise creators' solutions contained syntax or logic errors: "HAVING MAX(mSalary) wrong syntax."

	Chi^2	p-value	w
Difficulty	11.6	0^{***}	0.02932
Clarity	37.7	$<.001^{***}$	0.09518
Relevance	18.2	$<.001^{***}$	0.04593
Educational Value	0.19	0.91	0.00048

Table 1: Rating similarity for each criterion among student self assessment, peer assessment, and instructor assessment.

5.2 RQ2. Assessment Alignment

Despite the statistical significance, the differences between the three groups – student self-assessment, student peer assessment (average rating among all reviewers for each WE), and instructors – were negligible (Table 1). Although significant differences were observed in Difficulty, Clarity, and Relevance ($p < 0.001$) with Friedman's Chi-Squar Test [10], the Kendall's W coefficient [20] values for Difficulty, Clarity, and Relevance were negligible ($w < 0.1$).

We measured the reliability of student self and peer assessment ratings by their similarity to the instructor rating and the inter-rater agreement among students (Table 1). Figure 2 shows self, peer, and instructor ratings of the worked-out examples on each criterion.

Inter-rater agreement was found to be low among peer reviewers ($\alpha < 0.667$) across all four criteria. Further examination of the free-text comments indicated that the low agreement can be attributed to varying rating standards and the differing levels of evaluation efforts. Some students commented on the mistakes yet assigned high ratings in their evaluation. For example, one reviewer pointed out the wordiness of a question, stating that it made the instructions unclear, yet rated the relevance, clarity, and educational value of the WE as high. Another reviewer rated clarity as high while remarking, "this question does not make much sense to me." Furthermore, variations in effort could have also led to some disagreements. In one case, a reviewer did not identify any errors, while another reviewer noted several significant issues, including incorrect answers, failure to store data in variables, and omission of retrieving the origin. While individual peer reviewers may differ in their ratings due to subjective interpretations or varying effort levels, these individual variances tend to offset each other when averaged across a larger group of reviewers. Thus, despite low inter-rater reliability at the individual level, the central tendency (average) of multiple peer ratings aligns closely with the instructor's ratings.

6 Discussion

This research provides insights into the processes and challenges involved in engaging students in contextualizing computing concepts within their familiar scenarios. Our findings provide positive evidence for the feasibility of learnersourcing relatable learning materials in the form of worked-out examples. We also discovered that the instructor's examples significantly influenced many students' choice of contexts. Furthermore, our results suggest that more explicit instructions and scaffolding are necessary to help students provide essential details in their worked-out examples and ensure the correctness of their solutions. Below, we reflect on the initial learnersourcing design, identify three key challenges, and propose design implications to address these issues.

6.1 Challenges and Lessons Learned

Challenge 1. Ideate contexts from students' lived experiences beyond typical teaching and learning scenarios. The result reveals a strong inclination among students to rely on existing or popular examples provided by instructors (Subsection 5.1.1). While academic scenarios are inherently relatable to students, additional efforts are needed to elicit contexts that represent students' lived experiences beyond their learning environment to develop culturally responsive [34] learning materials. The findings highlight a challenge for students to recognize and identify their unique contexts (only two students achieved this). This scarcity may be linked to the stagnant culture in computing classrooms, where students usually focus on how to fit themselves into the prevailing computing culture [38], but are rarely asked to reflect on how computing integrates into their own contexts.

Challenge 2. Provide relevant information to explain the context and problem in the worked-out examples. While students successfully created worked-out examples (WEs) relevant to the course topics, challenges emerged in crafting clear problem descriptions and supplying sufficient data. These issues may cause other students to struggle to understand or engage with the WEs effectively. These quality concerns not only hinder the usability of learnersourced WEs for other students, but also limit the potential for student-created WEs to make a broader impact, gain recognition, and thereby increase student motivation and engagement with the learnersourcing approach.

Challenge 3. Develop step-by-step solutions that are clear, correct and executable. The occasional lack of details or errors in step-by-step solutions further hinders understanding and engagement with student-created WEs. In addition, explaining step-by-step solutions is critical for the effectiveness of WEs as a pedagogical method [31]. Notably, since students are in the process of learning computing concepts, it is reasonable to anticipate occasional misunderstandings and mistakes in their provided solutions. In a practical and scalable learnersourcing workflow, it is important to regard students as experts in training and accommodate varying levels of solution quality from them.

6.2 Design Implications

To address the challenges observed, and reflecting on the lessons learned, we draw six design implications for future improvement of learnersourcing workflows.

Implication 1. Foster Context Diversity through Examples. Effective learnersourcing activities should encourage students to reflect on their unique experiences and contexts, and provide more explicit guidance on how computing concepts can be applied in broader contexts. This could include allowing more time for the students to ideate new contexts, supplying additional training sessions or class activities, and providing examples that employ non-typical yet relatable scenarios. The examples could be prior successful worked-out examples created by other students and/or those demonstrating the instructors' own interests and experiences.

Implication 2. Enhance Context Integration through Smaller Steps. The key to contextualization is to integrate computing concepts meaningfully into learners' scenarios and vice versa. The learnersourcing workflow can be improved by further breaking down the steps of applying computational thinking in student scenarios and meaningfully applying computing concepts to solve problems. This could include scaffolding in formulating problem statements, defining deliverables, and developing test cases.

Implication 3. Provide Formatting Support. Learnersourcing workflows should be adapted to the learning materials being created and break the creation process down into smaller steps, with a template for the expected answers. For worked-out examples, this includes how the problem description should be formatted, how data should be provided, and how solutions should be presented. A preview of the resulting WE would help students identify missing information in their answers. This will not only streamline the task but also clarify the expectations for students.

Implication 4. Emphasize Clarity. Instructions provided to the students should emphasize the importance of precise and unambiguous language. Additional support, such as real-time clarity analysis as students draft their problem descriptions, could be helpful, but would require further research on how to balance support with distraction. More fine-grained rating rubrics would also help students reflect and refine the worked-out example during self-assessment.

Implication 5. Guided Data Preparation. The instructions should provide guidance and support to assist students in providing the data needed to engage with the worked-out examples they created. This involves prompting students to revisit their assumptions about the problem context, identify available data, and define distinct data fields. Providing support for generating meaningful dummy data is especially crucial for the database programming contexts.

Implication 6. Mitigate Influence of Incorrect Answers. To address usability issues due to the potential errors in students solutions, a possible improvement would be requesting descriptions of subgoals in each step. When utilizing worked-out examples created by students, peer assessment should promote critical thinking and encourage students to offer constructive suggestions for enhancing the quality of the examples created by their peers.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DUE-231568, CCF-2434429. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Solmaz Abdi, Hassan Khosravi, Shazia Sadiq, and Gianluca Demartini. 2021. Evaluating the Quality of Learning Resources: A Learnersourcing Approach. *IEEE Transactions on Learning Technologies* 14, 1 (2021), 81–92. https://doi.org/10.1109/TLT.2021.3058644

[2] Austin Cory Bart, Ryan Whitcomb, Dennis Kafura, Clifford A. Shaffer, and Eli Tilevich. 2017. Computing with CORGIS: Diverse, Real-World Datasets for Introductory Computing. *ACM Inroads* 8, 2 (mar 2017), 66–72. https://doi.org/10.1145/3095781.3017708

[3] Alessio Bellino, Valeria Herskovic, Michael Hund, and Jorge Munoz-Gama. 2021. A Real-world Approach to Motivate Students on the First Class of a Computer Science Course. *ACM Trans. Comput. Educ.* 21, 3, Article 22 (May 2021), 23 pages. https://doi.org/10.1145/3445982

[4] Dennis Bouvier, Ellie Lovellette, John Matta, Bedour Alshaigy, Brett A. Becker, Michelle Craig, Jana Jackova, Robert McCartney, Kate Sanders, and Mark Zarb. 2016. Novice Programmers and the Problem Description Effect. In *Proceedings of the 2016 ITiCSE Working Group Reports* (Arequipa, Peru) *(ITiCSE '16)*. Association for Computing Machinery, New York, NY, USA, 103–118. https://doi.org/10.1145/3024906.3024912

[5] Catherine C Chase, Doris B Chin, Marily A Oppezzo, and Daniel L Schwartz. 2009. Teachable agents and the protégé effect: Increasing the effort towards learning. *Journal of science education and technology* 18 (2009), 334–352.

[6] Thomas M Connolly, Carolyn E Begg, et al. 2006. A constructivist-based approach to teaching database analysis and design. *Journal of Information Systems Education* 17, 1 (2006), 43.

[7] Edward L Deci and Richard M Ryan. 2012. Self-determination theory. *Handbook of theories of social psychology* 1, 20 (2012), 416–436.

[8] Paul Denny, John Hamer, Andrew Luxton-Reilly, and Helen Purchase. 2008. PeerWise: Students Sharing Their Multiple Choice Questions. In *Proceedings of the Fourth International Workshop on Computing Education Research* (Sydney, Australia) *(ICER '08)*. Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/1404520.1404526

[9] Susan R. Fisk, Tiah Wingate, Lina Battestilli, and Kathryn T. Stolee. 2021. Increasing Women's Persistence in Computer Science by Decreasing Gendered Self-Assessments of Computing Ability. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 464–470. https://doi.org/10.1145/3430665.3456374

[10] Milton Friedman. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Amer. Statist. Assoc.* 32, 200 (1937), 675–701. https://doi.org/10.1080/01621459.1937.10503522 arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1937.10503522

[11] Stephen Gance. 2002. *Are constructivism and computer-based learning environments incompatible?* Ann Arbor, MI: MPublishing, University of Michigan Library.

[12] Elena L. Glassman, Aaron Lin, Carrie J. Cai, and Robert C. Miller. 2016. Learnersourcing Personalized Hints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing* (San Francisco, California, USA) *(CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 1626–1636. https://doi.org/10.1145/2818048.2820011

[13] Philip J. Guo, Julia M. Markel, and Xiong Zhang. 2020. Learnersourcing at Scale to Overcome Expert Blind Spots for Introductory Programming: A Three-Year Deployment Study on the Python Tutor Website. In *Proceedings of the Seventh ACM Conference on Learning @ Scale* (Virtual Event, USA) *(L@S '20)*. Association for Computing Machinery, New York, NY, USA, 301–304. https://doi.org/10.1145/3386527.3406733

[14] Mark Guzdial. 2010. Does Contextualized Computing Education Help? *ACM Inroads* 1, 4 (dec 2010), 4–6. https://doi.org/10.1145/1869746.1869747

[15] Wendy A Kellogg and Thomas Erickson. 2005. Supporting appropriation work with social translucence, collective sensemaking, and social scaffolding. *international reports on socio-informatics* 2, 2 (2005), 30–43.

[16] Hassan Khosravi, Kirsty Kitto, and Joseph Williams. 2019. RiPPLE: A Crowdsourced Adaptive Platform for Recommendation of Learning Activities. *Journal of Learning Analytics* 6 (12 2019). https://doi.org/10.18608/jla.2019.63.12

[17] Juho Kim et al. 2015. *Learnersourcing: improving learning with collective learner activity*. Ph. D. Dissertation. Massachusetts Institute of Technology.

[18] Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

[19] Eunbae Lee, Joseph A. Pate, and Deanna Cozart. 2015. Autonomy Support for Online Students. *TechTrends* 59 (7 2015), 54–61. Issue 4. https://doi.org/10.1007/s11528-015-0871-9

[20] Pierre Legendre. 2005. Species associations: The Kendall coefficient of concordance revisited. *J. Agri. Biol. Environ. Stat.* 10 (06 2005), 226–245. https://doi.org/10.1198/108571105X46642

[21] Christopher Mazzeo, Sara Y Rab, and Julian L Alssid. 2003. Building Bridges to College and Careers: Contextualized Basic Skills Programs at Community Colleges. (2003).

[22] John W McManus and Philip J Costello. 2019. Project based learning in computer science: a student and research advisor's perspective. *Journal of Computing Sciences in Colleges* 34, 3 (2019), 38–46.

[23] Paola Medel and Vahab Pournaghshband. 2017. Eliminating Gender Bias in Computer Science Education Materials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 411–416. https://doi.org/10.1145/3017680.3017794

[24] Briana B. Morrison, Lauren E. Margulieux, and Mark Guzdial. 2015. Subgoals, Context, and Worked Examples in Learning Computing Problem Solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (Omaha, Nebraska, USA) *(ICER '15)*. Association for Computing Machinery, New York, NY, USA, 21–29. https://doi.org/10.1145/2787622.2787733

[25] ANNEMARIE SULLIVAN PALINCSAR. 2012. Social constructivist perspectives on teaching and learning. *An introduction to Vygotsky* (2012), 290–319.

[26] Erika A Patall, Harris Cooper, and Susan R Wynn. 2010. The effectiveness and relative importance of choice in the classroom. *Journal of educational psychology* 102, 4 (2010), 896.

[27] Joan Peckham, Lisa L Harlow, David A Stuart, Barbara Silver, Helen Mederer, and Peter D Stephenson. 2007. Broadening participation in computing: issues and challenges. *ACM SIGCSE Bulletin* 39, 3 (2007), 9–13.

[28] Jean Piaget and Anita Tenzer. 1967. Six psychological studies. (1967).

[29] Nea Pirttinen, Arto Hellas, and Juho Leinonen. 2023. Experiences from Learnersourcing SQL Exercises: Do They Cover Course Topics and Do Students Use Them?. In *Proceedings of the 25th Australasian Computing Education Conference* (Melbourne, VIC, Australia) *(ACE '23)*. Association for Computing Machinery, New York, NY, USA, 123–131. https://doi.org/10.1145/3576123.3576137

[30] Johnmarshall Reeve and Hyungshim Jang. 2006. What teachers say and do to support students' autonomy during a learning activity. *Journal of Educational Psychology* 98 (2 2006), 209–218. Issue 1. https://doi.org/10.1037/0022-0663.98.1.209

[31] A Renkl. 2002. Worked-out examples: instructional explanations support learning by self-explanations. *Learning and Instruction* 12, 5 (2002), 529–556. https://doi.org/10.1016/S0959-4752(01)00030-5

[32] Mohammad Rob and Farhana Rob. 2018. Dilemma between constructivism and constructionism: Leading to the development of a teaching-learning framework for student engagement and learning. *Journal of International Education in Business* 11, 2 (2018), 273–290.

[33] Rod D Roscoe and Michelene TH Chi. 2007. Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of educational research* 77, 4 (2007), 534–574.

[34] Kimberly A Scott, Kimberly M Sheridan, and Kevin Clark. 2015. Culturally responsive computing: A theory revisited. *Learning, Media and Technology* 40, 4 (2015), 412–436.

[35] Thomas J Shuell. 1988. The role of the student in learning from instruction. *Contemporary Educational Psychology* 13, 3 (1988), 276–295. https://doi.org/10.1016/0361-476X(88)90027-6

[36] Anjali Singh, Christopher Brooks, and Shayan Doroudi. 2022. Learnersourcing in Theory and Practice: Synthesizing the Literature and Charting the Future. In *Proceedings of the Ninth ACM Conference on Learning @ Scale* (New York City, NY, USA) *(L@S '22)*. Association for Computing Machinery, New York, NY, USA, 234–245. https://doi.org/10.1145/3491140.3528277

[37] Minh Tran, Heather Killen, Jen Palmer, David Weintrop, and Diana Franklin. 2024. Harmonizing Scratch Encore: Scaffolding K-8 Teachers in Customizing Culturally Responsive Computing Materials. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1335–1341. https://doi.org/10.1145/3626252.3630756

[38] Mark Warschauer, Lonnie Turbee, and Bruce Roberts. 1996. Computer learning networks and student empowerment. *System* 24, 1 (1996), 1–14. https://doi.org/10.1016/0346-251X(95)00049-P

[39] Tasha Wyatt. 2015. Understanding the process of contextualization. *Multicultural Learning and Teaching* 10, 1 (2015), 111–132.