

<https://helda.helsinki.fi>

---

## Crowdsourcing in Computer Science Education

Pirttinen, Nea

ACM

2021-08-16

---

Pirttinen , N 2021 , Crowdsourcing in Computer Science Education . in ICER 2021: Proceedings of the 17th ACM Conference on International Computing Education Research . ACM , pp. 421-422 , ICER 2021: ACM Conference on International Computing Education Research , 16/08/2021 . <https://doi.org/10.1145/3446871.3469781>

---

<http://hdl.handle.net/10138/334411>

<https://doi.org/10.1145/3446871.3469781>

---

acceptedVersion

---

*Downloaded from Helda, University of Helsinki institutional repository.*

*This is an electronic reprint of the original article.*

*This reprint may differ from the original in pagination and typographic detail.*

*Please cite the original version.*

# Crowdsourcing in Computer Science Education

Nea Pirttinen  
University of Helsinki  
Helsinki, Finland  
nea.pirttinen@helsinki.fi

## ABSTRACT

Crowdsourcing is a method of collecting services, ideas, materials or other artefacts from a relatively large and open group of people. Crowdsourcing has been used in computer science education to alleviate the teachers' workload in creating course content, and as a learning and revision method for students through its use in educational systems. Tools that utilize crowdsourcing can act as a great way for students to further familiarize themselves with the course concepts, all while creating new content for their peers and future course iterations. In my research, I focus on investigating the effects of computing education systems that use crowdsourcing on students' learning, and the types of quality assurance methods required to use the artefacts students produce with these tools.

### ACM Reference Format:

Nea Pirttinen. 2021. Crowdsourcing in Computer Science Education. In *Proceedings of the 17th ACM Conference on International Computing Education Research (ICER 2021), August 16–19, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3446871.3469781>

## 1 INTRODUCTION

Crowdsourcing has great potential in alleviating lecturers' workload, especially when it comes to online courses. Creating new assignments for course iterations can be time-consuming, and having open source course materials and tools readily available could help with the workload. Also, with proper tools, it would be possible for students to participate in creation of the course assignments while learning the course topics at the same time. Crowdsourcing efforts that use the student population produce more content with less effort from the teachers, but require more filtering for quality assurance. Depending on the collected artefacts, this curation process can be at least partially automated, but the effectiveness of the automated quality assurance depends on the complexity of the crowdsourced content and the quality assurance methods itself.

My research extends the previous work regarding the use of CrowdSorcerer [6], and the quality and usability of the programming assignments created with the tool [7]. My research also strives to generalize the results and working solutions of existing educational tools that utilize crowdsourcing in computer science education. Automating crowdsourcing effectively requires investigating

technical solutions in addition to educational ones. Thus, my research will be multidisciplinary with focus on computer science, and additional interest in pedagogy and education.

## 2 RELATED WORK

Crowdsourcing is a method of obtaining services, ideas or goods from an open network of people, usually on the Internet. It has been used in a wide range of fields very successfully, computer science education included. For example, PeerWise [1], is a web-based tool for collecting multiple choice questions. All the student-created questions appear in the system as exercises other students can complete and review. Thus, PeerWise is largely a self-correcting system – multiple choice questions are simple enough for even novice students to check and correct, and the author of the exercise can edit their answers and explanations if the reviews point out any errors [1].

As an another example, CodeWrite [3] is a web-based tool used to crowdsource simple programming assignments from the student population. In CodeWrite, the students create an assignment description and fill out a method that works as a solution for their description, as well as create tests for their implementation. As with PeerWise, the assignments created with CodeWrite are published immediately for other students to complete and review. Similarly to CodeWrite, CrowdSorcerer [6] (described in more detail in Section 3) guides students through programming assignment and test case creation process.

Generally, educational crowdsourcing tools been deemed useful and well-liked by students. For example, student reviews of PeerWise have been positive, and the system has been described as a very good way of reviewing course topics [1, 2]. Similarly, based on student reports, CodeWrite helped the users to learn programming concepts, as they spent time investigating each other's solutions and comparing them to their own [3].

## 3 RESEARCH APPROACH

CrowdSorcerer [6] is a computer science education tool used for creating programming assignments. The tool can be embedded into online course materials, and the entire process of creating a full programming assignment is guided within the tool. The tool also has peer review functionality, in which students can examine and assess each other's assignments.

With CrowdSorcerer, students create the assignments based on the specifics given by the instructor: the condition can be, for example, to create an assignment that uses conditional clauses. The end result is a programming assignment with a handout, a model solution, a code template and test cases. The assignments are automatically tested for compilation errors and run through the user-generated tests to make sure the program functions as intended.

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ICER 2021, August 16–19, 2021, Virtual Event, USA

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8326-4/21/08.

<https://doi.org/10.1145/3446871.3469781>

Though my current work is mostly focused on CrowdSorcerer, my goal is to extend and generalize the research beyond one educational crowdsourcing system. Some of the proposed research questions include:

- **RQ1.** What types of crowdsourcing education tools exist?
- **RQ2.** How is the quality of crowdsourcing learning content evaluated?
- **RQ3.** How does using a crowdsourcing education tool affect students' learning?
- **RQ4.** Do different student subpopulations use crowdsourcing education tools differently?

For RQ1, the approach is to conduct a systematic literature review into the crowdsourcing education tools and practices that are used in the context of computer science education. RQ2 can be investigated by looking at the quality assurance methods in use at the moment, such as peer reviews and automated assessment of the crowdsourced content, and evaluating what methods are the most effective and reliable, using, for example, comparison of expert reviews and peer reviews. RQ3 can be answered through empirical research into what students learn through using these systems, and how much using the systems helps them. RQ4 can be approached by empirical research of the different student subpopulations, such as novice and experienced programmers.

### 3.1 Results

In an analysis of students' peer reviews to programming assignments created with CrowdSorcerer, we concluded that novices programmers can be as good reviewers as their peers with more programming experience in an introductory programming course's context [7]. This supports previous findings by Hamer et al. [4] in the context of comparing reviews by students and tutors for simple programming assignments. We also conducted a preliminary examination into the complexity of the crowdsourced programming assignments from an introductory programming course [8]. The results indicate that though the created assignments are mostly simple, at least at the beginning of the course, students are able to follow the given instructions and create programming assignments that meet the complexity expectations at the level of an introductory course. All these results are relevant for RQ2.

Using CrowdSorcerer to help students learn testing has also been investigated [5]. We found no statistical significance between the number of times students wrote tests with CrowdSorcerer and their tendency to answer testing-related questions in the course exam, or their performance in said testing-related exam questions. However, the study did highlight some issues related to teaching programming and testing, such as the increased complexity and overhead of creating both programming assignments and full tests cases with CrowdSorcerer. This study relates to RQ3.

### 3.2 Future work

There are no extensive literature reviews on the use of crowdsourcing in computer science education. This is the next step in my research, as a wide literature review into the use of crowdsourcing and crowdsourced content would be very beneficial to the field in general, as it would collect together the findings and current state of the research on this topic.

Besides the literature review, I plan conducting further studies into the reliability of peer reviews in the context of introductory programming courses and mostly novice reviewers. In my Master's thesis, I concentrated specifically on the quality of the peer reviews in order to inspect whether a laborious part of crowdsourcing, the quality assurance, can be at least partially conducted by peers. The next step would be replicating this study with multiple experts in order to verify the results of my Master's thesis.

All the created programming assignments are stored using the backend functionalities of CrowdSorcerer [6]. Lecturers and course assistants can gain access to these assignments, and filter them according to various parameters. Thus, it would be possible to put these assignments into a suitable database from which they can be readily used in various contexts, such as drill practice or quiz exercises. Assignments are labeled with tags (topic of the assignment, such as recursion), and through peer reviews, we can assess the assignments perceived difficulty level and various other parameters. An open, global assignment bank would provide simple programming assignments to be used in any programming course. As the format of the programming assignments is generic, this material bank could be used widely after its publication, easing the workload of teachers in various levels of education. The use of these simple programming assignments is not restricted to tertiary education, but could also benefit, for example, middle and high school teachers who may not be as well-versed in teaching programming.

## REFERENCES

- [1] Paul Denny, Andrew Luxton-Reilly, and John Hamer. 2008. The PeerWise System of Student Contributed Assessment Questions. In *Proceedings of the Tenth Conference on Australasian Computing Education - Volume 78* (Wollongong, NSW, Australia) (ACE '08). Australian Computer Society, Inc., AUS, 69–74.
- [2] Paul Denny, Andrew Luxton-Reilly, and John Hamer. 2008. Student Use of the PeerWise System. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education* (Madrid, Spain) (ITiCSE '08). ACM, New York, NY, USA, 73–77. <https://doi.org/10.1145/1384271.1384293>
- [3] Paul Denny, Andrew Luxton-Reilly, Ewan Tempero, and Jacob Hendrickx. 2011. CodeWrite: Supporting Student-Driven Practice of Java. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) (SIGCSE '11). Association for Computing Machinery, New York, NY, USA, 471–476. <https://doi.org/10.1145/1953163.1953299>
- [4] John Hamer, Helen C. Purchase, Paul Denny, and Andrew Luxton-Reilly. 2009. Quality of Peer Assessment in CS1. In *Proceedings of the Fifth International Workshop on Computing Education Research* (Berkeley, CA, USA) (ICER '09). Association for Computing Machinery, New York, NY, USA, 27–36. <https://doi.org/10.1145/1584322.1584327>
- [5] Vilma Kangas, Nea Pirttinen, Henrik Nygren, Juho Leinonen, and Arto Hellas. 2019. Does Creating Programming Assignments with Tests Lead to Improved Performance in Writing Unit Tests?. In *Proceedings of the ACM Conference on Global Computing Education* (Chengdu, Sichuan, China) (CompEd '19). Association for Computing Machinery, New York, NY, USA, 106–112. <https://doi.org/10.1145/3300115.3309516>
- [6] Nea Pirttinen, Vilma Kangas, Irene Nikkarinen, Henrik Nygren, Juho Leinonen, and Arto Hellas. 2018. Crowdsourcing Programming Assignments with CrowdSorcerer. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) (ITiCSE 2018). Association for Computing Machinery, New York, NY, USA, 326–331. <https://doi.org/10.1145/3197091.3197117>
- [7] Nea Pirttinen, Vilma Kangas, Henrik Nygren, Juho Leinonen, and Arto Hellas. 2018. Analysis of Students' Peer Reviews to Crowdsourced Programming Assignments. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (Koli, Finland) (Koli Calling '18). Association for Computing Machinery, New York, NY, USA, Article 21, 5 pages. <https://doi.org/10.1145/3279720.3279741>
- [8] Nea Pirttinen and Juho Leinonen. 2021. Exploring the Complexity of Crowdsourced Programming Assignments. In *Seventh SPLICE Workshop at SIGCSE 2021 "CS Education Infrastructure for All III: From Ideas to Practice"*. <https://cssplice.github.io/SIGCSE21Workshop.html>