Leveraging Community Software in CS Education to Avoid Reinventing the Wheel

Jeremiah Blanchard*
Dept. of Engineering Education
University of Florida
Gainesville, Florida, USA
jjb@eng.ufl.edu

John R. Hott*
Dept. of Computer Science
University of Virginia
Charlottesville, Virginia, USA
jrhott@virginia.edu

Vincent Berry
Polytech Engineering School
LIRMM - Univ Montpellier, CNRS
Montpellier, France
vincent.berry@umontpellier.fr

Rebecca Carroll
Dept. of Computer Science
Full Sail University
Winter Park, Fl, USA
rebeccac@fullsail.edu

Bob Edmison
Dept. of Computer Science
Virginia Tech
Blacksburg, Virginia, USA
bedmison@vt.edu

Richard Glassey
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden
glassey@kth.se

Oscar Karnalim†
School of Information and Physical Sciences
University of Newcastle
Callaghan, NSW, Australia
oscar.karnalim@uon.edu.au

Brian Plancher
Dept. of Computer Science
Barnard College, Columbia University
New York City, NY, USA
bplancher@barnard.edu

Seán Russell
School of Computer Science
University College Dublin
Dublin, Ireland
sean.russell@ucd.ie

ABSTRACT

Historically, computing instructors and researchers have developed a wide variety of tools to support teaching and educational research, including exam and code testing suites and data collection solutions. Many are then community or individually maintained. However, these tools often find limited adoption beyond their creators. As a result, it is common for many of the same functionalities to be re-implemented by different instructional groups within the CS Education community. We hypothesize that this is due in part to accessibility, discoverability, and adaptability challenges, among others. Further, instructors often face institutional barriers to deployment, which can include hesitance of institutions to utilize community developed solutions that often lack a centralized authority.

This working group will explore what solutions are currently available, what instructors need, and reasons behind the above-mentioned phenomenon. This will be accomplished via a literature review and survey to identify the tools that have been developed by the community; the solutions that are currently available and in use by instructors; what features are needed moving forward for classroom and research use; what support for extensions is needed to support further CS Education research; and what institutional challenges instructors and researchers are currently facing or have faced in the past in developing, deploying or otherwise using community software solutions. Finally, the working group will identify factors that limit adoption of solutions and ways to integrate and improve the accessibility, discoverability, and dissemination of existing community projects, as well as manage and overcome institutional challenges.

CCS CONCEPTS

• **Social and professional topics** → **Student assessment**; • **Software and its engineering** → *Software libraries and repositories*; **Open source model**.

KEYWORDS

educational tools, community software

ACM Reference Format:
Jeremiah Blanchard, John R. Hott, Vincent Berry, Rebecca Carroll, Bob Edmison, Richard Glassey, Oscar Karnalim, Brian Plancher, and Seán Russell. 2022. Leveraging Community Software in CS Education to Avoid Reinventing the Wheel. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol 2 (ITiCSE 2022), July 8–13, 2022, Dublin, Ireland*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3502717.3532169

1 MOTIVATION AND GOALS

Practically from the birth of computing disciplines, instructors and researchers have built, used, and published software for community use to assist one another with student assessment and research in computing coursework [3–5]. At first, source was shared through journals and books [3]. As the Computing Education Research

*Working group leader
†Also with Maranatha Christian University.

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).
ITiCSE 2022, July 8–13, 2022, Dublin, Ireland
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9200-6/22/07.
https://doi.org/10.1145/3502717.3532169

(CER) community grew, so too did the avenues for distribution, with the rise of networking and eventually ubiquitous access and the unprecedented capacity for sharing source through the Internet. However, as with many industries, the easy publication and lack of centralized listing and filtering mechanisms resulted in the difficulty in discovering many tools, a problem that persists to this day [7]. Even when tools are discovered, however, instructors and researchers may find that institutions are hesitant to endorse the use of solutions that do not have a centralized authority, as is common in community-driven projects. By comparison, centralized solutions offered by large technology firms are more easily discovered through marketing and name-recognition. These firms also offer a clear, recognized authority that can be held accountable, assuaging many institutional concerns during risk assessment processes. These solutions almost always come at a cost, though, whether directly via fees, indirectly via surrender of some degree privacy, or both [8].

Some instructors and researchers find the costs associated with such platforms burdensome and/or objectionable, particularly when fees are ultimately passed on to students, and they may be an insurmountable hurdle to students and faculty from socio-economically disadvantaged populations or regions. As a result, instructors and researchers often resort to developing in-house solutions, essentially reinventing the wheel by generation and institution [3–6]. However, there are some community-driven, non-profit solutions that have developed within the CER community that have managed to achieve success, both in terms of discoverability and institutional deployment [1, 2]. Likewise, many Open Source Software (OSS) projects have found success and wide utilization. There are lessons that the CER community can learn from these examples that could be applied more broadly to help instructors and researchers build on existing tools and frameworks.

This working group will explore what instructional and research solutions are currently available, those that are in demand, reasons for the challenges to wider adoption and collaboration, and potential avenues to support community software development and adoption in the future by identifying and building on successes in the CER and OSS communities. Specifically, the goals of this working group will be to work with a diverse, multinational group of researchers to:

(a) Review literature, identifying existing community software in CS Education and its functionality, barriers to discoverability and adoption of such software, and successful models to support community development;
(b) Draft, refine, and deploy an international survey to educators and education researchers soliciting feedback on experiences, challenges, and initiatives related to community software development and use in classrooms and research, as well as the current needs of the computing education community;
(c) Analyze results, in concert with reviewed literature, to identify ways to integrate and improve the accessibility and discoverability of existing community projects, as well as manage and overcome institutional challenges; and
(d) Disseminate results of the survey and findings from analysis to the international community of computing educators and researchers.
(e) Suggest directions for a combined effort of the community to expose and disseminate new solutions, as well as reduce development to adoption time.

2 METHODS

The working group will conduct a comprehensive review of existing literature and develop, deploy, and analyze the results of a community survey.

2.1 Literature Review

The comprehensive literature review will identify available tools, historical challenges to adoption, and successful models of dissemination for community-developed and maintained software. We will incorporate works that document the following:

(1) Community projects (and related course materials) in service of computing education teaching and research,
(2) Successful community projects that have been widely adopted,
(3) Case studies related to the deployment of community software, and
(4) Institutional standards or procedures that may facilitate or inhibit adoption of tools.

2.2 Community Survey

A community survey will be developed by members of the group in parallel and complementary to the literature review. The survey will include questions on the following topics:

(1) Tools that have been developed by participants;
(2) Tools in use by participants (community or other);
(3) Satisfaction with current tools, and reasoning for such;
(4) Features and/or tools participants would like to see;
(5) Institutional barriers to community software deployment, development, and/or maintenance; and
(6) Non-institutional barriers to community software deployment, development, and/or maintenance.

REFERENCES

[1] Martin Dougiamas and Peter C Taylor. 2002. Interpretive Analysis of an Internet-based Course Constructed using a New Courseware Tool called Moodle. In *2nd Conference of HERDSA (The Higher Education Research and Development Society of Australasia)*. 7–10.
[2] Stephen H. Edwards. 2014. Work-in-Progress: Program Grading and Feedback Generation with Web-CAT. In *First ACM Conference on Learning @ Scale Conference*. ACM, 215–216. https://doi.org/10.1145/2556325.2567888
[3] George E. Forsythe and Niklaus Wirth. 1965. Automatic Grading Programs. *Commun. ACM* 8, 5 (may 1965), 275–278. https://doi.org/10.1145/364914.364937
[4] Sheung-Lun Hung, Iam-For Kwok, and Raymond Chan. 1993. Automatic Programming Assessment. *Computers & Education* 20, 2 (1993), 183–190. https://doi.org/10.1016/0360-1315(93)90086-X
[5] Julia Isong. 2001. Developing an Automated Program Checkers. *Journal of Computing Sciences in Colleges* 16, 3 (2001), 218–224.
[6] David Jackson. 1996. A Software System for Grading Student Computer Programs. *Computers & Education* 27, 3-4 (1996), 171–180. https://doi.org/10.1016/S0360-1315(96)00025-5
[7] Fenwick McKelvey and Robert Hunt. 2019. Discoverability: Toward a Definition of Content Discovery Through Platforms. *Social Media+ Society* 5, 1 (2019). https://doi.org/10.1177%2F2056305118819188
[8] Panagiotis Papadopoulos, Nicolas Kourtellis, Pablo Rodriguez Rodriguez, and Nikolaos Laoutaris. 2017. If You Are Not Paying for It, You Are the Product: How Much Do Advertisers Pay to Reach You?. In *Internet Measurement Conference*. ACM, 142–156. https://doi.org/10.1145/3131365.3131397