

# Civic Engagement Across the Computing Curriculum

Mark Goadrich (Moderator)  
Hendrix College  
Conway, AR, USA  
goadrich@hendrix.edu

Michael Goldweber  
Xavier University  
Cincinnati, OH, USA  
goldweber@xavier.edu

Matthew Jadud  
Bates College  
Lewiston, ME, USA  
matthew.c@jadud.com

S. Monisha Pulimood  
The College of New Jersey  
Ewing, NJ, USA  
pulimood@tcnj.edu

Samuel A. Rebelsky  
Grinnell College  
Grinnell, IA, USA  
rebelsky@grinnell.edu

## ABSTRACT

As outlined in the ACM Computer Science Curricula 2013 Guidelines section on Social Issues and Professional Practice, "Students must also be exposed to the larger societal context of computing to develop an understanding of the relevant social [and] ethical ... issues." [1] In this panel, we demonstrate diverse approaches used to achieve this goal with respect to civic engagement. Drawing from experiences with non-major, introductory computing, mobile applications, software engineering, and interdisciplinary courses, we discuss how to move beyond surface-level discussions of ethical case studies toward an integration of civic engagement activities and personal reflection into standard computing curriculum.

## CCS CONCEPTS

• **Applied computing** → **Computing in government; Education;**

## KEYWORDS

Civic Engagement, Computing for Social Good, Social Issues

## 1 SUMMARY

Many standard computing courses can be enhanced with opportunities for civic engagement. Our experiences show that with careful planning and support, incorporating civic engagement can increase student motivation and success, and open new pathways for underrepresented groups in computing. Our intended panel audience is computing educators seeking advice and direction on how to begin implementing civic engagement in their courses. We are also targeting educators looking to share their own civic engagement experiences and build community.

## 2 PANEL STRUCTURE

After a very brief introduction from each panelist (5 minutes total), the moderator will ask the panel to comment on the following questions, with the goal of drawing out common themes and connections

among the panelists on their experiences with civic engagement and computing curriculum.

- How do you involve civic engagement in your courses?
- How do you assess student success?
- What external resources helped or barriers hindered you?
- What lessons have you learned?

We recognize that our positions cannot cover the full spectrum of civic engagement experiences. Thus, each of these questions will also be posed to the audience to enlarge the conversation. Discussion on each question will be limited to 10 minutes, leaving 30 minutes for audience questions and participation.

## 3 POSITION STATEMENTS

### 3.1 Mark Goadrich

In fall of 2017, I enhanced our CS1 course at Hendrix with civic engagement modules. The first project in our introductory computer science course consisted of students coding a simple question/answer situation. I required their program help the user understand a civic process, such as applying to be a U.S. citizen, or determining which permits were required for renovating a home in a historic district, and provided sample output for one such situation. With a civic direction and purpose for the project, we found that students spent much more time on this project than in previous semesters. The students became very invested and engaged in getting all of the details correct, often going above and beyond the requirements to earn full credit. Second, I created a lab exercise on civic hacking to visualize freely-available data from our local government website. Given the GPS locations for local educational institutions and for tornado sirens inside city limits, students were tasked to determine if there was sufficient overlap between the effective siren range and the schools. In a follow-up survey, I was encouraged that my students were able to extrapolate their experience to other civic hacking activities, such as investigating pedestrian safety, or relationships between crime and police station locations.

### 3.2 Michael Goldweber

In the Computing for Social Good in Education (CSG-Ed) literature there is a four-level taxonomy of how "green" an assignment or exercise can be. These levels range from level 1: redefining an existing project with a CSG-Ed narrative, up to level 4: a real world problem brought by or solved for real stakeholders. In the fall of

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).

SIGCSE '19, February 27–March 2, 2019, Minneapolis, MN, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5890-3/19/02.

<https://doi.org/10.1145/3287324.3287335>

2017 I offered an experimental course: “Open-Ended Problem Solving” with the goal of undertaking a level 4 project. After consulting with various community partners, I decided to focus on the current opioid epidemic. After spending a crash course on the opioid epidemic through guest lectures by community partners, students, in hackathon fashion engaged in an ideation phase. Once there were  $n$  reasonably developed ideas, there was a voting phase to select the 3 most popular projects. Finally, students, through preference voting, were assigned to project teams. Each team, composed of 6-8 students, self-organized to tackle their selected problems. Focusing on two week sprints, all teams adopted the agile approach. All in all there were 5 sprints: an insufficient number to bring any of the apps to completion. While I was pleased with the educational experience—students were very successful in their interactions with community partners and in identifying new technologies, algorithms, and concepts they then went on to master—the unrealized potential of bringing these apps to market left all feeling unsatisfied. This is, however, a common pitfall with level 4 projects. Unexpectedly, I received a \$25k grant to hire students to complete 2 of the apps. Six students were hired, none of whom were enrolled in the original class; testing both the quality of documentation and students’ ability to perform a project handoff. By the conclusion of the summer, both apps had received approval from our primary community partners and are awaiting Google and Apple approval.

### 3.3 Matthew Jadud

At Bates College, during our 5-week intensive “short term,” where students enroll in one course, we (Jadud and co-educator Hamish Cameron) offered *Community Engaged Computing*, an introductory course with 14 first- through final-year students. The course met four mornings per week (2hrs) and two afternoons (3hrs), plus community engagement time on Fridays (2hrs). The course involved learning the fundamentals of programming in App Inventor, a community-engaged design process, and two series of readings: one on the history of immigration in Lewiston, Maine (*Making Refuge: Somali Bantu Refugees and Lewiston, Maine* by Catherine Besteman) and a series of readings on creativity and creative process. The community partner in this project was an affinity group of middle-school aged Somali and Somali Bantu girls who met weekly after school for community and mutual support. The team of Bates college students who chose to work with this group (three women and one man) first developed a short lesson for the students to introduce them to programming in App Inventor, and then used the images and resources from this project to develop a replicable protocol for turning these images and stories into a mural. This work was carried out in close collaboration with Erica Mott (artist, choreographer) and Justice Roe (muralist) in Chicago, Ill. The product of the collaboration was then hung in the Hyde Park Art Center, and will later be presented in venues local to Lewiston.

### 3.4 S. Monisha Pulimood

I teach Software Engineering, a Computer Science major requirement, and have found that traditional projects that appeal to some students do not resonate with others. Inspired by research that shows that students, particularly those from underrepresented groups, are more engaged by projects that address a social need,

I explored collaborations with other disciplines and community partners. Supported by NSF Award No. 1141170, a colleague from Journalism and I developed and tested a curricular model that enables students with diverse perspectives and disciplinary backgrounds to learn how to collaborate and integrate concepts from their respective fields to develop computational solutions for complex real-world problems. This includes coordinated, but separately taught, courses and collaboration with a community partner. The pilot consisted of a multi-semester collaboration between students in computer science, journalism and interactive multimedia classes who worked in partnership with Habitat for Humanity, to create a web-based application to identify pollutants in properties being considered for redevelopment. Interdisciplinary collaborative classes emphasize problem-solving in a gender-neutral, culturally and ethnically diverse community, and provide an engaging learning environment in which students solve problems together with their peers from other disciplines. Moreover, students are motivated to explore social justice issues more deeply, and understand how they can use their disciplinary backgrounds to address a community-identified need. My classes continue to collaborate with other local non-profits, such as the local food bank. The software engineering course is now designated as meeting the Advanced Community Engaged Learning requirement for students.

### 3.5 Samuel A. Rebelsky

For the past four years, the emphasis of our software design course has been multi-semester projects in which small teams (4-6 student) build non-mission-critical software for community nonprofits. Students have developed a wide variety of projects, including a text notification system for a local preschool, a grant management system for the umbrella for local nonprofits, and a directory tailored toward a local retirement home. Our partner nonprofits report that they can rarely afford the commercial equivalents to the software the students develop, which gives the partners patience when projects take more time than predicted. Students report that knowing the software they write will actually make a difference in the local community helps motivate and incentivize them. Although most projects take a year or more to develop, students typically work on a project for only a semester, giving them a more realistic software development experience, one in which they regularly join projects in progress. As they work with non-technical partners and grow in their understanding of the broader context of the projects they are developing, students also build their “soft” skills.

## ACKNOWLEDGMENTS

Mark Goadrich’s work was supported by Project Pericles. S. M. Pulimood’s work was partially supported by the National Science Foundation (NSF) under Grant No. DUE-1141170. 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 NSF.

## REFERENCES

- [1] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA. 999133.