A Scaffolding-based Approach for Addressing Challenges of Service Learning Adoption

Stan Kurkovsky
kurkovsky@ccsu.edu
Central Connecticut State University
New Britain, CT, USA

Mikey Goldweber
mikeyg@denison.edu
Denison University
Granville, OH, USA

Chad A. Williams
cwilliams@ccsu.edu
Central Connecticut State University
New Britain, CT, USA

Nathan Sommer
sommern1@xavier.edu
Xavier University
Cincinnati, OH, USA

Abstract

Recent publications strongly support refocusing undergraduate CS education toward competency-based learning. This shift places increased responsibility on departments to prepare responsible computing practitioners who appreciate how computing is inextricably intertwined with society. It emphasizes that content coverage is less important than authentic experiences that develop both ethical decision-making and industry-desired technical skills. One time-tested successful strategy that helps meet these goals is community-based service learning (CBSL). However, while CBSL has a strong track record, service learning can be challenging to implement and may not always guarantee successful student experiences.

A key factor associated with CBSL's success or failure is the management of the community partner relationship. This includes initial project vetting, setting partner expectations, the role of the partner as a participant in the students' education, and final project hand-off. We introduce Scaffolded Projects for the Social Good (SPSG), a framework based on the software studio model that guides CBSL adopters through all stages of a CBSL experience. The SPSG framework pays particular attention to what are considered the most vexing aspects of CBSL: project scoping and skill matching, managing project timelines that extend beyond a single term, community partner engagement and relationship management, and project handoff and maintenance.

Preliminary results from the adoption of the SPSG framework demonstrate that students were able to iteratively improve their competencies throughout the semester as a result of the regular formative feedback enabled by the SPSG framework.

ACM Reference Format:
Stan Kurkovsky, Mikey Goldweber, Chad A. Williams, and Nathan Sommer. 2025. A Scaffolding-based Approach for Addressing Challenges of Service Learning Adoption. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3641555.3705119

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0532-8/25/02
https://doi.org/10.1145/3641555.3705119

Further Information

Despite a broad range of CBSL benefits (e.g. [1, 3, 5]), adopting CBSL presents significant challenges. **Identification of an appropriate non-profit community partner** is challenging because many partners lack the technical expertise, experience, or understanding of the educational requirements of the course to be an *effective partner* without additional assistance. **Identification of an appropriate project** is challenging because the partner may overestimate the scope of what a team of undergraduate students can accomplish over one or two semesters. **Project time frame** rarely aligns with what can be accomplished in a single semester. **Institutional support** may be needed to adopt a competency-based model in the course supporting CBSL projects. Adopting any new pedagogical approach often requires a substantial **time commitment** as instructors will need to spend time performing different, potentially unfamiliar tasks.

Scaffolded Projects for the Social Good (SPSG)[1] is a joint effort by the authors to address the issues related to incorporating CBSL into the computing curriculum and helping focus computing education on student competencies. It was conceived with the idea of formalizing a studio model [2] in order to bring CBSL within reach of individual instructors and programs, regardless of the institution size or the level of support infrastructure that may be available at their disposal. In SPSG, each project may span one or more semesters when different teams may be contributing to it. To that end, SPSG places a very strong emphasis on continuous formative feedback provided to student teams by the course instructor and the project partner. This is achieved through a multitude of low-stakes deliverables spread throughout the semester that also facilitate knowledge transfer among the teammates and across different teams working on the project during a different semester.

SPSG projects are grounded in agile principles, emphasizing short iterations, customer engagement, and flexibility to adapt to changing requirements and challenges. Each semester includes four phases: inception, elaboration, development, and transition, each tailored to the project's scope. The **inception** phase precedes the semester, where the instructor collaborates with the project partner to assess feasibility, align with student teams' capabilities, define scope and duration, identify points of contact and methods of communication, and establish expected outcomes. This results in a

[1] https://spsg-hub.github.io

standardized project proposal delivered to the student team(s). Student teams are formed at the beginning of the **elaboration** phase. They collaborate closely with the project partner to understand and define project requirements. For new projects, teams conduct multiple rounds of customer interviews to outline the requirements, which are then translated into user stories to form the initial product backlog. The **development** phase follows agile methodology, organized into two-week sprints. At the start of each sprint, the team selects user stories from the product backlog for implementation. Due to students' limited availability (up to 10 hours per week), daily scrum meetings are replaced with a weekly scrum held during a scheduled class session. In-class self-reflection at sprint retrospectives allows teams to evaluate their achievements, discuss lessons learned, and adjust both the product backlog to reflect changes in project requirements and the team's workflow to optimize their effectiveness. The **transition** phase occurs during the final week of the semester and focuses on knowledge transfer. Several outcomes are possible during this phase: the project continues into the next semester with the same team(s); the project continues into the next semester with at least one new team; or the project is completed, with no active teams involved aside from routine maintenance.

SPSG provides a robust scaffolding of student deliverables spread across the semester that ensures continued student engagement with the course project and provides the course instructor with ample opportunities to evaluate student work and provide student teams with formative feedback. These deliverables include a team agreement, project requirements outline and several versions of the product backlog, as well as deployment and transition documentation. For every development sprint, teams produce a sprint backlog and a sprint report. Regular scrums and in-class retrospectives allow for feedback between student teams, increased accountability among the students, and a heightened sense of community.

SPSG shares similarities with Humanitarian Free and Open Source Software (HFOSS) projects [6]. While both approaches aim to facilitate service learning, SPSG distinguishes itself by having a significantly lower adoption threshold, addressing several shortcomings of HFOSS, and seamlessly integrating both formative and summative assessments of student outcomes. Unlike HFOSS, SPSG projects work directly with project partners representing community organizations, non-profits, or similar entities, thus putting many other stakeholders typically involved in an HFOSS project (e.g. project maintainers) out of the decision-making loop. Human interactions are key in SPSG emphasizing the need for students to work directly with project partners. This collaboration develops professional competencies through real-world experience, providing opportunities to understand the impact of their software solutions on the constituents of the non-profit or community organization.

The SPSG framework was used with several recent projects undertaken by four teams consisting of 4-5 seniors in the Senior Project course at a four-year medium-sized primarily nonresidential public institution. Throughout the projects, the SPSG framework helped expose students to many aspects concerning the societal impacts of computing, communication, teamwork, and ethics. Witnessing their project partners' dedication, the teams gained a deeper appreciation for **social responsibility**. All project partners were very passionate about the causes they and their projects served. Although this observation is purely anecdotal, we feel that this enthusiasm was contagious and that it imparted a lasting difference on the students. Two of these projects focused on **developing computing solutions for populations in need**. For many students, it was eye-opening to see how technology we take for granted can significantly impact others' lives. Both projects addressed resource scarcity, including power supply, network connectivity, and digital infrastructure.

To independently evaluate student competencies, all four teams presented their projects at a semi-annual Senior Project Showcase. A panel comprised of Industrial Advisory Board members and alumni evaluated the projects using a standardized rubric aimed to assess student attainment of learning outcomes aligned with professional competencies including technical depth, communication and presentation, and professionalism.

The SPSG framework is currently being piloted and refined at two institutions: a medium-sized public university and a small private liberal arts college. We are planning to expand our community of practice and work with additional institutions to pilot the framework in a more diverse array of contexts. Feedback from schools with different student demographics, program specifics, and institutional contexts will allow us to further refine the framework and account for assumptions we have made based solely on our own experiences. Having exercised SPSG ourselves, we are confident that the framework's structure is helpful in navigating the instructional logistics of implementing CBSL, and from anecdotal evidence through working with students we believe participation in CBSL projects has a positive impact on students beyond improving their technical and professional skills. Desiring stronger evidence of the latter, we are in the early stages of conducting a longitudinal study to measure the impact of student participation in CBSL projects on their formation and development of professional dispositions [4] and on their attitudes toward social responsibility [7].

Acknowledgments

This work was supported in part by NSF awards DUE-2315322 and DUE-2315323.

References

[1] Alexander W Astin, Lori J Vogelgesang, Elaine K Ikeda, and Jennifer A Yee. 2000. How service learning affects students. *Higher Education* (2000).

[2] Christopher N. Bull and Jon Whittle. 2014. Supporting Reflective Practice in Software Engineering Education through a Studio-Based Approach. *IEEE Software* 31, 4 (2014), 44–50. https://doi.org/10.1109/MS.2014.52

[3] Randy W Connolly. 2012. Is there service in computing service learning?. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 337–342.

[4] Natalie Kiesler, Amruth N. Kumar, Bonnie K. MacKellar, Renée McCauley, Mihaela Sabin, and John Impagliazzo. 2024. Students' Perceptions of Behaviors Associated with Professional Dispositions in Computing Education. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1* (Milan, Italy) *(ITiCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 353–359. https://doi.org/10.1145/3649217.3653566

[5] Barbara E Moely and Vincent Ilustre. 2014. The Impact of Service-Learning Course Characteristics on University Students' Learning Outcomes. *Michigan Journal of Community Service Learning* 21, 1 (2014), 5–16.

[6] Lori Postner, Darci Burdge, Stoney Jackson, Heidi Ellis, George Hislop, and Sean Goggins. 2015. Using Humanitarian Free and Open Source Software (HFOSS) to Introduce Computing for the Social Good. *SIGCAS Comput. Soc.* 45, 2 (July 2015), 35–35. https://doi.org/10.1145/2809957.2809967

[7] Daniel S Schiff, Emma Logevall, Jason Borenstein, Wendy Newstetter, Colin Potts, and Ellen Zegura. 2021. Linking personal and professional social responsibility development to microethics and macroethics: Observations from early undergraduate education. *Journal of Engineering Education* 110, 1 (2021), 70–91.