

# Starting a Civic Engagement Capstone: An Experience Report\*

Mark Hills and James B. Fenwick, Jr.  
Department of Computer Science  
Appalachian State University  
Boone, NC  
`{hillsma, fenwickjb}@appstate.edu`

## Abstract

In this paper, we reflect on the first semester of a Computer Science capstone course focused on civic engagement projects. This was offered as an alternate section of an existing capstone, with multiple student teams working with a local non-profit. We describe how the student work was structured during the semester, as well as the activities leading up to the semester and occurring immediately after the semester. We also reflect on what worked well, and discuss potential changes in future semesters. We believe this will be helpful to other faculty that are either starting a new capstone course, or modifying an existing capstone course, that will engage community partners with student teams.

## 1 Introduction

Starting in Spring 2024, the Computer Science Department at Appalachian State University introduced a special section of its capstone course focused on *civic engagement* projects—projects with a client (in our case, a local non-profit) selected from the surrounding community. Civic engagement is an informal approach to experiential, service learning. Service learning has its philosophical and pedagogical roots in the work of Dewey [5], Freire [6], and

---

\*Copyright ©2022 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

Kolb [8]. A key tenet is abandoning the idea of education being a “banking process” that fills students with knowledge and facts. Instead, education is viewed as a reciprocal relationship where teachers also learn and learners also teach. This approach can support deep learning as students work up Bloom’s taxonomy [1] synthesizing classroom knowledge in practical and local settings. Students benefit in multiple and varied ways, including academic knowledge and skills, interpersonal communication skills, and increased self confidence [13].

In this paper, we reflect on our initial experience in adding a civic engagement section of our capstone course. Many students enter college with idealistic ambitions of changing the world. Sometimes these ambitions can become lost in the curricular routine of assignments and short-lived, academic projects. Capstone courses offer students an opportunity to revisit those early ambitions with more skill and knowledge, while adding a service-learning component affords students an easy entry to working on a fulfilling project with a positive, immediate, and direct impact on an observable community organization. We start in Section 2 by describing the preexisting capstone course. Section 3 then provides details about the civic engagement section of the capstone, including what was done before the semester began, student and faculty tasks during the semester, and student reflections on their work once the semester concluded. Section 4 summarizes the results of an informal survey sent to a CS education mailing list to solicit information about similar capstones in other programs, while Section 5 highlights some existing research on capstone courses in computer science and software engineering (including some highlighted by responses to our survey). Section 6 concludes by reflecting on what worked well and what modifications may be made in future semesters to overcome difficulties encountered during the Spring 2024 semester. Our hope is that this paper will be helpful to faculty in other programs that are considering starting such a capstone or that have recently started such a capstone.

## 2 Background

The current capstone course, CS 4800, is generally taken by undergraduate students in their final year in the program. The catalog description of the course is: "The senior capstone project provides the student an independent and collaborative software development experience with a significant project. The course introduces aspects of project management, requirements analysis, and the software lifecycle, but will primarily be concerned with the practical integration of core theories, practices, and ethics of the discipline. Writing and speaking communication skills are reinforced." Students enrolling in the course must have senior standing, and must have already taken the Software Engineering and Data Structures courses, the latter of which is also a gateway

to many of our upper-level electives. Typical enrollment is around 20 students per section. The course meets for 3 hours once per week. During this time, students generally give status updates, present the current status of their work, and work on their projects. Although students are expected to also work on the projects outside of class, the class session provides dedicated project time each week, which can be especially helpful if they are stuck and need assistance from the instructor. Having the full team together in person also helps re-center the team as students often have challenges in aligning their out-of-class schedules.

Students in a typical section of CS 4800 work on personal, self-selected projects. Most students work on individual projects, but occasionally students with shared interests work on teams of 2 or (more rarely) 3 students. Early in the term, students propose a project idea which, if approved, they work on over the course of the semester. The technologies are up to the students, and vary based on student interests and the type of project. Core courses in the program use Java, C/C++ and SQL, with other common languages including Python, JavaScript, Swift, and Kotlin (depending on which other courses students take). While many students use these languages, and the techniques used in earlier courses, some students use the capstone as an opportunity to learn a new language or platform. Several checkpoints, including a final report and a final project presentation, occur throughout the semester. Common examples of projects from past semesters include games, personal assistants, and utilities, with console, GUI, web, and mobile applications all being common choices. The typical capstone project starts and ends during the academic term. In some cases students may begin with an existing personal project, and the capstone would continue building on what they had already developed.

### **3 The Civic Engagement Capstone**

In this section, we provide details about the Spring 2024 civic engagement capstone section. We look at activities before the start of the semester, including the inception of the civic engagement section; describe the activities of the capstone during the semester; and look at the results of the semester, including client and student feedback.

#### **3.1 Before the Semester**

During the Fall 2023 semester, we were approached by an employee of the non-profit that we worked with in the capstone. They had received a grant to expand their services and wanted technology assistance to develop a client portal. Shortly after this, the university Center for Excellence in Teaching and Learning for Student Success sent out a Request for Proposals, focused on integrating civic engagement into a course. Inspired by this RFP, we decided to add

a special section of our capstone course focused on civic engagement projects. Although the immediate goal was to teach a special section in Spring 2024, the longer-term goal was to continue this work into future semesters, providing a mechanism for students to get involved in team projects with community partners that often have a lifetime of more than a single semester. In contrast to the projects described in Section 2, these projects would not be created by the students, but would instead be based on the needs of our community partners, and would (except in unusual circumstances) all be team projects. We applied for the RFP and were awarded funds to help cover small stipends for the faculty involved, to purchase technology books for the students, and to travel to speak with other faculty working on similar capstone courses.

### 3.2 During the Semester

The capstone course met a total of 14 times over the course of the Spring 2024 semester. Faculty met with two possible clients before the start of the semester: a community partner (the local nonprofit mentioned above, which is the focus of this experience report) and a startup. During week 1, the instructor presented initial details about the clients and their needs, then worked with the students to develop questions for each client for a planned client visit. This occurred during week 2, with each client visiting the course, presenting further details about their needs, and answering questions from the authors and the students. After the clients departed, the instructor and the students decomposed the overall work for each client into multiple projects: five for the community partner, and two for the startup. The projects for the nonprofit were an API backend server, a web client, and three mobile clients: an Android client, an iOS client, and a cross-platform client (added to explore the design space of possible mobile solutions). Students then ranked their interest in each project in a survey form and were assigned to teams based on their responses.

Week 3 then started the actual work on the projects. 17 students were divided into 5 teams for the nonprofit: 3 students on each mobile project, and 4 on the API server and web client projects. Each student project was organized according to Scrum principles, with two week sprints, self-directed student teams, and the instructor acting as the Scrum Master, helping students follow best practices. During the first week, each team spend class time to get organized for the rest of the project. The instructor created teams for each student team on GitHub. Each team then created one or more GitHub repositories for code collaboration, and created GitHub project boards to track their work. They also started planning the actual work of the project, creating initial issues to represent team tasks and making initial decisions on technology (e.g., the technology stack they would use) and teamwork (e.g., collaboration workflows for effectively using GitHub). This was the start of Sprint 1.

Each week, before class, each team submitted a single status report. These reports were submitted by creating a document on a shared folder, with the goal of providing a history for future semesters. Each class meeting started with a Scrum standup meeting, where each team could describe what they had accomplished over the past week and what issues were preventing them from making progress. Class meetings then ended with details of what the team planned to accomplish over the next week. During class time, teams could work together on their projects and meet with the clients. Every other week, the teams would present the current state of their project, with the goal of quickly moving towards a minimal viable product that would provide a basis for future work. Meetings with the community partner were either held over Zoom or were asynchronous, with the teams coordinating to email collections of questions instead of each emailing the client individually. Meetings with the startup were held either in-person or over Zoom. Student teams communicated using a Discord server created by the instructor or using other preferred communication mediums (e.g., text messages, other Discord servers, email).

In the week before the final class meeting, the student teams focused on ensuring the project was in a good state to be picked up by the next teams in the fall. The semester then ended with the submission of a project report (built over the course of the semester, with the instructor giving feedback at multiple checkpoints) and final project presentations by all teams, which were attended by the clients. Each student team also met individually with the instructor for a more detailed project demonstration and in-depth discussion. In general, students made good progress on individual projects, but work on linking the projects (e.g., the website to the backend API server) has mainly been deferred to Fall 2024.

### 3.3 After the Semester

At the end of the semester, a student survey with the following questions was sent out to all students in the civic engagement section of the capstone:

- Q1** What teamwork skills did you improve, or learn, during the project?
- Q2** How effective do you believe your team was? (1 to 5 scale, 1 is dysfunctional, 2 is ineffective but functional, 3 is sufficient, 4 is effective, 5 is highly effective)
- Q3** What do you believe could have been done better to help your team work together more effectively?
- Q4** What technical skills did you improve, or learn, during the project?

- Q5** What do you believe could have been done better to help you learn new technical skills during the project?
- Q6** How much communication was there with the other teams? (1 to 5 scale, 1 is Not Enough, 3 is Just Right, 5 is Too Much)
- Q7** How much communication was there with the client? (Same scale as Q6)
- Q8** Did communication with the client come at the right times? (1 to 5 scale, 1 is at the wrong times, 5 is at just the right times)
- Q9** How did you communicate with the client? (Email, Text, Telephone, Discord, Zoom, In Person, or Other, multiple answers allowed)
- Q10** What could be done in the future to improve communication with external clients? What worked well, and what could be done differently?
- Q11** Feel free to share any other thoughts about your experience working on your project.

Students could only respond once to the survey, but student emails were not revealed as part of the collected data so responses were anonymous.

Of the 23 students in the class, 7 responded to the survey. Note that some of these responses may have been from the students on the projects with the startup—while still useful, they do not bear directly on the experience of the students working with the community partner. The students gave their teams an average effectiveness of 3.43 (question 2). Looking at communication, respondents gave average ratings of 2.43 for question 6 (communication between teams, with 3 being “just right”), 1.71 for question 7 (communication with the client, same scale), and 2.57 for question 8. All students selected “email” as one communication option, with all but one also selecting Zoom. One common theme was the importance of communication with the client, and the need for more regular, scheduled times to meet. Comments included “I think it would be very helpful if the client worked closer with the groups”, “Meeting with them in person more often”, and “I think having more face to face or zoom calls would be helpful”. We discuss this further in Section 6.

We also met with the community partner for a debriefing session. In general, they were excited about the progress the students made during the semester. They were especially happy with the coordination of the teams around visual elements, such as the use of the same fonts and color schemes that they use in their materials. They had several ideas for additional features, and committed to staying involved during the Fall 2024 semester.

## 4 An Informal Capstone Survey

To identify other programs that offer project-based courses with external clients, especially those that have projects that extend across semesters, we sent a survey to the ACM SIGCSE (Special Interest Group in Computer Science Education) mailing list. Note that the survey was intentionally kept quite short and informal—our goal was to make it easy to respond, giving us a chance to gather some initial information which could then be enriched with future conversations. This survey included the following questions:

- Q1** Does your undergraduate degree require students to take a capstone course near the end of their time in your program? Note that this may be called “Capstone”, but could also be a project-based course under another name.
- Q2** If you answered “Yes” to Question 1: Do you allow your students to work on projects with community partners and other outside individuals and/or organizations?
- Q3** If you answered “Yes” to Question 2: Do you have projects that last for more than a single term/semester?
- Q4** Would you be willing to either have a short conversation over Zoom with us or answer additional questions about your capstone course?
- Q5** Do you have any additional thoughts to share about capstone courses, working with external partners/on external projects, or multi-term projects?

We received a total of 16 responses to the survey. Of these, 14 stated that their programs offer some kind of capstone experience for their students, with 1 additional program offering a course that, while not a capstone, has students work on a significant software project. Ten of the programs with a capstone course require students to take this course. All 15 of the programs with a capstone or capstone-like course allow engagement with community partners, while 13 of these programs have projects which can last more than one semester, some by design (e.g., a two-semester capstone sequence), some by choice of the student (e.g., a student can retake the course for additional credit and can continue the same project). Seven of the responses stated that a project may be worked on by different student teams over time.

While we cannot generalize these results to programs outside of those of the respondents, the results have given us more information about how the responding programs run similar capstone courses. We plan to carry out a more rigorous study in the future, across a broader collection of programs, that will hopefully yield more detailed information.

## 5 Related Work

As described by Coyle et al. [3] and by Oakes et al. [12], Engineering Projects in Community Service (EPICS) is a model for integrating community engagement throughout the curriculum. Students participate in projects that span multiple years, with project teams made up of students from multiple disciplines, including Computer Science. Started at Purdue [16], EPICS has expanded to multiple other universities, including Princeton, Ohio State University, and Arizona State University [17]. Linos et al. [10] describe adapting the EPICS model to a new Software Engineering program in a liberal arts setting, including a discussion of ongoing challenges around topics such as the continuity of projects and effective project management. One significant difference between the work on EPICS and the work we describe above is that we are just focused on the capstone course at this point, so students are all senior-level students that are just taking the course for a semester.

Davis and Rebelsky [4] describe the design of a software design course which includes multi-semester projects with community partners. Kurkovsky et al. [9] introduced a service-learning framework named Scaffolded Projects for the Social Good, or SPSG. This uses a studio-based approach for student projects with external partners, and is similarly run during development using agile techniques with two-week sprints. For continuity, students can work on the same projects across an undergraduate software engineering and a senior project or capstone class, and spend the last week of the semester focused on knowledge transfer activities. Braught et al. [2] describe multiple models for introducing H/FOSS (Free and Open Source, potentially with Humanitarian goals) projects into the curriculum, with models varying on the organization of the capstone course and related courses, such as courses in software engineering. All of these models appear to have goals similar to ours, but with potentially different course sequences, so we plan to follow up with the authors.

Dugan [7] conducted a survey of existing literature on undergraduate capstone courses in Computer Science. This survey organized the literature around both course and project goals, including topics such as which software process models were used and which course topics were covered. Similarly, Tenhunen et al. [14] performed a systematic literature review of software engineering capstone courses, where they categorized courses on characteristics such as team size, type of client, and duration.

## 6 Final Reflections and Future Work

Several important points came out during the capstone. These are described below as communication, team support, and infrastructure.



## 6.1 Communication

The most important challenge in Spring 2024 was with communication, including between teammates on a single team, between student teams, and with the client. Anecdotally, communication on individual teams went well, especially since students had time each week to work together in class, but there were times during the semester when teams struggled with communication specifically about tasks to do outside of class. This seemed to get better during the semester as students become more familiar with one another and with the task tracking tools on GitHub. Providing online training on how to use these tools, and introducing them in more detail in class, may help the next teams collaborate more effectively from the start. For communication between teams, the students that responded to the survey though there could be more communication than there was (2.43, with 3.0 being ideal). This coming semester, class time for these inter-team meetings will be explicitly added to the schedule.

The main communication struggles were with the community partner. Expectations with the client need to be more clearly set up front to ensure they are available at the right times, and students need more guidance about when to use the different communication channels they have available with the client. Communication challenges, especially limited face to face communication opportunities, were the main pain point mentioned by the students. An important goal for Fall 2024 will be to find the right balance for both the students and the clients. We plan to coordinate with the clients before the start of the semester to ensure regular meeting dates are already on the calendar before students again start to work on the projects.

A related point is that non-technical clients may need additional scaffolding around how to give effective feedback to student teams during these meetings. It can be challenging for clients that have never been involved in a software project to precisely formulate their needs, comment on completed work, or select from different alternative feature options. The students found it effective to offer options from a limited collection of alternatives, since this helped the clients to know what was possible and what tradeoffs were available. It may make sense to create more guidance for the clients in the future, and (an idea from our meeting with an instructor at another university) to separate out presentations into technical presentations (for the instructors and the other students) and non-technical presentations (focused specifically on the clients).

## 6.2 Team Support

Although all the teams worked well together this past semester, we could imagine conflicts arising on future teams. To help with this, we plan to use a tool like CATME [11, 15] to allow students to give one another peer feedback dur-

ing the semester and to raise concerns with the instructor. Student teams may also need more hands-on support in using Scrum, breaking down work into reasonable pieces, and focusing on high-priority issues for the clients, although this went reasonably well in Spring 2024.

### 6.3 Infrastructure

Finally, more focus needs to be placed on infrastructure requirements for student projects. Some of this is non-technical infrastructure: although triggered by a client approaching us, to keep this section active we will most likely also need to recruit and manage clients from the community. Looking at technical infrastructure, to fully deploy the software the student teams created for our community partner, we would need servers to host the website and API server, and would also need accounts with Google and Apple to publish apps to the app stores. While university-provided versions of these resources work during development, at some point client-owned versions of these resources will be needed. If students are working on multiple teams with dependencies between the teams (e.g., an API server with related clients, each developed by different teams), determining where to host projects needed by other teams needs to be done early in development to prevent problems later on, where student projects have inter-project dependencies that block progress.

### 6.4 The Future

In this paper, we presented our initial experience with adding a civic engagement section of our current capstone course. The student teams were able to make good progress on their projects viewed as individual projects, but the work of joining these projects into a single portal will mainly fall to the Fall 2024 cohort. We've identified communication as the most important issue to focus on going forward, but also pointed out other potential challenges above.

All artifacts created by the student teams, such as weekly status reports, the final project reports, and all presentations, are available to future teams on a shared Google Drive, while all code and project boards are hosted on GitHub. Fall 2024 will be the test of whether this is sufficient for onboarding new student teams that will have the (to students, often quite novel) challenge of reading, understanding, and modifying code that they did not create themselves. We look forward to seeing what challenges—hopefully foreseen, potentially unforeseen—arise.

**Acknowledgments:** This work was funded in part by a Civic Engagement Innovation Grant from the Center for Excellence in Teaching and Learning for Student Success at Appalachian State University.

## References

- [1] P. Armstrong, *Bloom's Taxonomy*. 2010. URL: <https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/>.
- [2] Grant Braught et al. "A Multi-Institutional Perspective on H/FOSS Projects in the Computing Curriculum". In: *ACM Trans. Comput. Educ.* 18.2 (July 2018). DOI: 10.1145/3145476.
- [3] Edward J. Coyle, Henry G. Dietz, and Leah H. Jamieson. "Long-Term Community Service Projects in the Purdue Engineering Curriculum". In: *Proceedings of the 1996 American Society for Engineering Education Conference*. 1996.
- [4] Janet Davis and Samuel A. Rebelsky. "Developing Soft and Technical Skills Through Multi-Semester, Remotely Mentored, Community-Service Projects". In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE '19. Association for Computing Machinery, 2019, pp. 29–35. DOI: 10.1145/3287324.3287508.
- [5] J. Dewey. *Experience and Education*. New York: Touchstone, 1938.
- [6] P. Freire. *Pedagogy of the Oppressed*. New York: Continuum, 1970.
- [7] Robert F. Dugan Jr. "A survey of computer science capstone course literature". In: *Computer Science Education* 21.3 (2011), pp. 201–267. DOI: 10.1080/08993408.2011.606118.
- [8] D. A. Kolb. "Experiential Learning: Experience as the Source of Learning and Development". In: *Journal of Business Ethics* 1.15 (1984), pp. 45–57.
- [9] Stan Kurkovsky et al. "Scaffolded Projects for the Social Good: A Strategy for Deploying Studio Model in CS Education". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. SIGCSE 2024. Association for Computing Machinery, 2024, pp. 1706–1707. DOI: 10.1145/3626253.3635487.
- [10] Panagiotis K. Linos, Stephanie Herman, and Julie Lally. "A Service-Learning Program for Computer Science and Software Engineering". In: *SIGCSE Bull.* 35.3 (June 2003), pp. 30–34. DOI: 10.1145/961290.961523.
- [11] Misty L. Loughry, Matthew W. Ohland, and D. DeWayne Moore. "Development of a Theory-Based Assessment of Team Member Effectiveness". In: *Educational and Psychological Measurement* 67.3 (2007), pp. 505–524. DOI: 10.1177/0013164406292085.

- [12] William C. Oakes, Edward J. Coyle, and Leah H. Jamieson. “EPICS: A Model of Service-Learning in an Engineering Curriculum”. In: *Proceedings of the 2000 American Society for Engineering Education Conference*. 2000.
- [13] Suzanne Savanick et al. *Service Learning*. URL: <https://serc.carleton.edu/introgeo/service/index.html>.
- [14] Saara Tenhunen et al. “A systematic literature review of capstone courses in software engineering”. In: *Information and Software Technology* 159 (2023), p. 107191. DOI: 10.1016/j.infsof.2023.107191.
- [15] Purdue University. *CATME: Smarter Teamwork*. <https://catme.org/>.
- [16] Purdue University. *Welcome to EPICS*. <https://engineering.purdue.edu/EPICS>.
- [17] Wikipedia. *Engineering Projects In Community Service*. [https://en.wikipedia.org/wiki/Engineering\\_Projects\\_In\\_Community\\_Service](https://en.wikipedia.org/wiki/Engineering_Projects_In_Community_Service).