

## **Board 4: Computers in Education Division: Partnering to Develop Educational Software Applications: A Four-year Retrospective Study**

**Mr. David Reeping, Virginia Tech**

David Reeping is a doctoral candidate in Engineering Education at Virginia Tech and is an NSF Graduate Research Fellow. He received his B.S. in Engineering Education with a Mathematics minor from Ohio Northern University. He was a Choose Ohio First scholar inducted during the 2012-2013 school year as a promising teacher candidate in STEM. David was the recipient of the Remsburg Creativity Award for 2013 and the DeBow Freed Award for outstanding leadership as an undergraduate student (sophomore) in 2014. He is also a member of the mathematics, education, and engineering honor societies: Kappa Mu Epsilon, Kappa Delta Pi, and Tau Beta Pi respectively. He has extensive experience in curriculum development in K-12 and creates material for the Technology Student Association's annual TEAMS competition. David has co-authored two texts related to engineering, *Principles of Applied Engineering* for Pearson-Prentice Hall and *Introductory Engineering Mathematics* for Momentum Press.

His research interests include: model/method transferability, threshold concepts to inform curriculum development, information asymmetry in higher education processes (e.g., course articulation), and issues in first year engineering.



## Partnering to Develop Educational Software Applications: A Four-year Retrospective Study

**Mr. David Reeping, Virginia Tech**

David Reeping is a graduate student pursuing a Ph.D. in Engineering Education at Virginia Tech and is an NSF Graduate Research Fellow. He received his B.S. in Engineering Education with a Mathematics minor from Ohio Northern University. He was a Choose Ohio First scholar inducted during the 2012-2013 school year as a promising teacher candidate in STEM. David was the recipient of the Remsburg Creativity Award for 2013 and the DeBow Freed Award for outstanding leadership as an undergraduate student (sophomore) in 2014. He is also a member of the mathematics, education, and engineering honor societies: Kappa Mu Epsilon, Kappa Delta Pi, and Tau Beta Pi respectively. He has extensive experience in curriculum development in K-12 and creates material for the Technology Student Association's annual TEAMS competition. David has co-authored two texts related to engineering, *Principles of Applied Engineering* for Pearson-Prentice Hall and *Introductory Engineering Mathematics* for Momentum Press.

His research interests include: model/method transferability, threshold concepts to inform curriculum development, information asymmetry in higher education processes (e.g., course articulation), and issues in first year engineering.

**Dr. John K. Estell, Ohio Northern University**

Dr. John K Estell is Professor of Computer Engineering and Computer Science at Ohio Northern University, providing instruction primarily in the areas of introductory computer programming and first-year engineering. He has been on the faculty of the Electrical & Computer Engineering and Computer Science Department since 2001, and served as department chair from 2001-2010. He received a B.S.C.S.E. degree from The University of Toledo and the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Urbana-Champaign. Dr. Estell is a Fellow of ASEE, a Senior Member of IEEE, and a member of ACM, Tau Beta Pi, Eta Kappa Nu, Phi Kappa Phi, and Upsilon Pi Epsilon.

Dr. Estell is active in the assessment community with his work in streamlining and standardizing the outcomes assessment process, and has been an invited presenter at the ABET Symposium. He is also active within the engineering education community, having served ASEE as an officer in the Computers in Education and First-Year Programs Divisions; he and his co-authors have received multiple Best Paper awards at the ASEE Annual Conference. His current research includes examining the nature of constraints in engineering design and providing service learning opportunities for first-year programming students through various K-12 educational activities. Dr. Estell is a Member-at-Large of the Executive Committee for the Computing Accreditation Commission of ABET, and also serves as a program evaluator for the Engineering Accreditation Commission. He is also a founding member and serves as Vice President of The Pledge of the Computing Professional, an organization dedicated to the promotion of ethics in the computing professions through a standardized rite-of-passage ceremony.

# **Partnering to Develop Educational Software Applications: A Four-Year Retrospective Study**

## **Introduction**

Several years ago, a project was added to the first-year programming sequence at Ohio Northern University that focused on developing educational software applications. The intent was to show how computing and engineering professionals could make a positive difference in the lives of others, even if limited to working from behind a computer screen. However, the project was largely self-directed by the students, as each team picked the subject for a provided grade level and then wrote a software application for it. While the project allowed students to display technical competency, it embraced a person-centric view towards learning as an internal process.

In his seminal work, Kolb discusses the characteristics associated with the concept of experiential learning [1], where thoughts and ideas are not fixed, but are formed and reformed through continuous processes grounded in experience and involving transactions between the learner and the environment. Kolb's experiential learning model is based on such transactions, where knowledge is created through the transformation of external, real-world experiences. One form of incorporating real-world experiences involves service learning, where students are actively involved with community-based experiences meeting one or more societal needs. It was posited that repurposing the software application project to incorporate service learning concepts could better serve the students involved. Therefore, in 2014 the project was revised in the spirit of experiential learning to meet the needs of clients from the field of education. The project has since expanded to redirect the narrative structure of the first-year programming sequence and serves as the sequence's culminating experience. As the project enters its fifth year of use, it is now an appropriate time to conduct a retrospective analysis and share both the salient elements of what has enabled its persistence and the perceived benefits of adopting the service-based orientation. The hypothesis for this preliminary longitudinal research study asserts that first-year programming students can gain some of the benefits associated with service learning, as measured by a positive shift in mindset toward engaging in service, through building client-based relationships with members of educational learning communities.

## **Literature Review**

Service learning involves combining academic learning with community service in such a way that the service performed is integral to achieving the academic goals of a course. Service learning projects are well documented within higher education, including within engineering via established programs as EPICS [2, 3] and Engineers Without Borders [4, 5], and are implemented with the goal of developing a positive mindset toward service learning. Similarly, the concept of incorporating the entrepreneurial mindset [6, 7] into engineering education has resulted in project-based service learning experiences. For example, students enrolled in a course at Villanova are first given the cultural context of a developing country and then live in one of that country's rural communities during spring break [8]. The exposure to and immersion within the new culture allows for the first-hand discovery of potential projects that can exhibit both technical merit and potential business opportunities.

Relatively few examples of service learning experiences were found within the computing literature, primarily due to three perceived shortcomings addressed in [9]. First, to perform useful service, students should be experienced, which thereby limits such offerings to just upper-level courses. Second, a useful project typically involves a complex problem. Due to the “hidden” nature of software implementation to the end users, an unmotivated or weaker programmer could inadvertently cause harm that goes undetected until well after the fact. Third, software often requires maintenance. Unlike a commercial entity, students are transitory, thereby leaving the end user adrift if a problem occurs or if a modification is needed. Amongst those efforts reported on regarding applying service learning in computer science [10, 11, 12, 13, 14], some have involved experiences being conducted in the first year of study. Examples include creating animations for a community music outreach program [15], teaching basic computer skills to prison inmates [16], performing usability and accessibility audits of non-profit websites [9], conducting computing lessons in an after-school program at a local middle school [17], and applying the entrepreneurial mindset to the service learning experience [18].

## **Project Design**

To formalize the design of the project, this section briefly describes the learning objectives, the participants involved, and the course context.

### *Learning Objectives:*

After engaging in the project, the students will be able to:

- (1) Collaborate in a team setting to develop GUI-based software applications, and
- (2) Understand the motivations and perspectives of stakeholders.

Although not a learning objective for the course, but a broader goal of the college, the motivation of introducing this project was to cultivate a service-oriented mindset in the students; hence the following hypothesis was posed:

- (H) First-year programming students can gain some of the benefits associated with service learning, in terms of an increased positive attitude toward service, through building client-based relationships with members of various educational learning communities.

### *Participants and Course Context:*

The first set of participants are the students enrolled in the Programming 2 course, which is the second course in the introductory programming sequence at Ohio Northern University. Both introductory courses have three 50-minute lectures and one 175-minute laboratory session weekly during the semester, with the laboratory being used to reinforce just-covered lecture material. The first course uses C++ to provide experience in breaking problems down into functional units involving sequence, selection, and iteration; whereas the second course uses Java to explore the event-driven graphical user interface paradigm through object-oriented programming practices. The students enrolled in these courses are primarily computer engineering and computer science majors.

The second set of participants are the project clients. Over the four years covered by this study, four different sets of clients were used. In 2014, the clients were engineering education students developing materials in support of educational outreach activities. The software applications

were designed to augment lesson plans developed as part of a series of teacher workshops that the students helped to deliver in the Dominican Republic. Due to the departure of the instructor leading the then-annual trips, in 2015 the focus shifted to having ASEE student chapter members serve as clients in support of their engineering outreach activities in local middle school classrooms. For 2016, the instructor collaborated with the Ohio Northern University Education Placement Office to recruit local fourth and fifth grade teachers as clients. In this case, student teams worked on augmenting lesson plans to be delivered later that academic year. Unfortunately, working with practicing educators had a major drawback: the lack of direct interaction between clients and student teams due to the teachers' busy work schedule. To address the shortcoming, clients have been recruited since 2017 from a group of education majors enrolled in a gamification course offered by one of the authors, with prerequisites set such that they already had developed lesson plans in hand from prior coursework.

## Implementation

### *Theoretical Framework:*

The current version of the project was implemented as a cornerstone project (a term commonly used to refer to a culminating first-year engineering design experience) in 2014 within the second semester Programming 2 course of Ohio Northern University's first-year programming sequence. To ground the project in a pedagogical framework, this section will outline the theoretical underpinnings of the project design.

As mentioned in the Introduction, the Kolb Cycle of Experiential Learning, illustrated in Figure 1, was used to help organize the series of cornerstone activities into a cyclic pattern of experiences and reflections. The cycle was augmented by Greenaway's Active Reviewing Cycle, a model which provides a different way to examine experiential learning [19]. The keywords from this cycle are shown within parentheses in Figure 1.

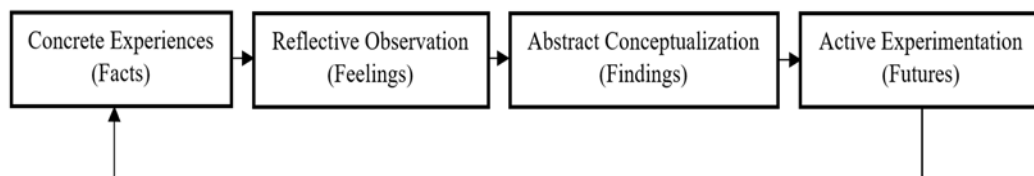


FIGURE 1. KOLB EXPERIENTIAL LEARNING CYCLE WITH GREENWAY'S KEYWORDS

The *concrete experience* stage is used to engage students in performing some sort of activity where they apply their ideas and skills. Experiences from activities generate *facts* – the events, moments, and details associated with the activity. Next, the *reflective observation* stage encourages students to reflect on their experiences through mechanisms such as self-evaluation, peer discussion, and instructor feedback. Reflections generate *feelings*, an assessment of the experience from various modes of input. During the *abstract conceptualization* stage, students integrate their observations by generating *findings*, answering the “why did that happen” question and thereby adding to one’s learning by synthesizing their perceptions and reflections into theory revisions. Finally, through *active experimentation* students apply the revised theories to generate *futures*, allowing them to make decisions and to solve problems by preparing to test new ideas and skills.

### *Structure of Project:*

The overall structure of the project is predicated on all clients having an established lesson plan that is aligned either to Common Core [20] or Next Generation Science Standards [21], and that applications are developed for students primarily within the range of elementary (K-5) or middle school (6-8) grade levels. Through the first constraint, the project receives buy-in from educators due to their need for standards-aligned materials. With the second constraint, the college students designing the software applications are sufficiently removed from that prior stage of their lives that they now must make a concerted effort to understand the needs of the noticeably younger students who will be using their software. The point concerning user needs is of importance, as it is unfortunately too often the case that software developers design by envisioning how they would use the software instead of realizing that “you are not your user” [22]. Each client’s lesson plan is summarized into a description that concisely addresses what students should know, how students are to reach this goal, and why students are learning the content. The lesson plan summaries are compiled into a single document that is sent to all programming teams. Each team indicates their top choices and submits them to the instructor, who then makes the final selections and assigns teams to projects.

Once assignments are made, each team is tasked with envisioning two possible applications by explaining how each app supports the lesson plan, outlining what the app will do for the students, and hand-sketching what a sample screen might look like when the app is in use. The teams then meet with their client to discuss the sketches, after which they use the client’s feedback to narrow their focus to a single app for further development. In lieu of going out and interviewing grade school students, personas are used to help the programming students discern and thereby address the typical wants, needs, and fears of their app’s end users. At this point, the team fills out a Project Proposal form that is reviewed by both the client and the instructor. A Proposal Evaluation Rubric, constructed using the single-point format [23, 24], is used to provide formative feedback by indicating whether the team meets the stated expectations, or is either above or below, for each assessed dimension. Based on this feedback, the team develops a Minimum Viable Product (MVP) based on the Lean Startup methodology [25] for discerning the key items to implement that allows them to show their client a prototype of the application.

At this stage the teams are required to start submitting weekly progress reports and obtaining feedback on the current state of their app, beginning with the MVP implementation, via a Software Application Evaluation (SAE) Rubric, during a three-week implementation period. Additionally, teams are provided with periodic feedback regarding how they conduct their client meetings through use of the Student-Client Interaction Rubric [26]. In the penultimate week of the term, the apps undergo a Critical Design Review (CDR) through the holding of a Software Application Fair. All teams are present in the lab, and each app is reviewed during one-on-one sessions from judges drawn from both Education and Engineering faculty, along with advisory board members and programming laboratory assistants. These reviews also use the SAE Rubric for providing written feedback to the teams. Teams are required to record and submit a summary of the oral comments made by all judges, and then use the MVP process to prioritize app changes to be implemented in the week between the CDR and final app submission. As part of the final submission, an oral presentation is made to the class, a set of reflective questions are individually answered, and peer evaluations are performed.

## Sample Student Work

To better illustrate what student teams have proven themselves capable of through the cornerstone design project, two software applications are briefly reviewed here; example screenshots from each application are presented in Figure 2.



FIGURE 2. SCREEN IMAGES FROM WATER FILTRATION AND LANGUAGE CONVENTIONS APPLICATIONS.

### *Water Filtration:*

This application was written in support of workshops conducted in the Dominican Republic in 2014; accordingly, the user interface is in Spanish. The application presents the user with the ability to experiment with a five-stage water filtration system that is fabricated from a set of 15 different filtration materials, presented graphically underneath the *Materiales* header. Through experimentation the user can determine which filtration materials to use and in what order for creating an optimal design by dragging a selected material with the cursor to one of the five filtration stages (*Etapa*) provided underneath the *Niveles de Filtro* (filter levels) header. By clicking on the *Presentar* (submit) button, the user is provided with a rating for the design.

### *Language Conventions:*

This application was written in 2017 to augment a lesson plan for fourth grade language conventions, where students will be able to use relative pronouns and relative adverbs, form and use the progressive verb tenses, use modal auxiliaries to convey various conditions, order adjectives within sentences according to conventional patterns, form and use prepositional phrases, produce complete sentences, and correctly use frequently confused words. The programming team created a Mad Libs-type game, where the user fills in the blank with an appropriate part of speech, to review the nouns, verbs, adjectives, and homophones aspects of language conventions. Different short stories were created for each character, where the use of correct words control progress through the game. Students are kept engaged by revealing only one aspect of the Mad Lib at a time, and as they progress students can unlock accessories such as hats for their avatar to wear. Three levels of hints are employed to assist struggling students, and mini-games with language convention related themes are used as rewards.

## Assessment Design

After four years of offering the cornerstone project and revising the approach accordingly, the assessment design has taken on a structure built upon a culture of formative assessment, as opposed to summative. Summative assessment is much like a post-mortem, collected at the end of a unit, course, or program; and is intended to measure competency [27] and to comply with accountability standards [28]. Formative assessments, on the other hand, are more like routine physicals or checkups; such assessments occur during the unit, course, or program and are subsequently used to improve the students' learning experience before completion [27, 28]. There are two levels to the assessment design, student and project – to be described separately.

### *Student Level:*

As student teams are working with external clients, the use of formative assessment techniques allows the teams to receive non-judgmental ungraded feedback that can then be used to improve the product. As mentioned in the description of the implementation, the single-point format was used in developing a set of rubrics for formative assessment. A distinguishing feature of the single-point rubric is the elimination of multiple performance criteria level descriptions in favor of specifying, as a “single point,” the description outlining the expected performance under that criterion. There is no defined upper bound on performance, which can limit creativity, or lower bound, which can minimize expectations. For each criterion, the team can be evaluated as either being above, at, or below expectations; blank spaces on the instrument are liberally provided “to provide specific written feedback on various aspects of students' work that will help them know how to improve” [23]. The student level assessment is thus intimately tied with Kolb's Experiential Learning Cycle, as each new experience during the project allots for a period of assessment, reflection, reconceptualization, and experimentation with the students' new findings.

### *Project Level:*

To provide an evaluation of the project with respect to developing a service-oriented mindset, the Community Service Attitude Scale (CSAS) survey [29, 30] was the primary assessment instrument selected for this research. Consisting of 46 7-point Likert scale questions, the CSAS survey was developed to measure 10 aspects of college students' attitudes toward community service. First, the scale considers the student's *awareness* of community needs, the perception of *actions* that can meet the need, and the perception of one's own *ability* to help. Second, the scale considers one's sense of *connectedness* to one's own community that motivates helping, one's sense that personal or situational *norms* obligate one to help, and one's sense of *empathy* for those in need. Moreover, both the *costs* and the *benefits* to oneself of helping are included with respect to the student's beliefs about the *seriousness* of the consequences of not helping others, and one's *intentions* to engage in community service. This instrument has been evaluated for validity and reliability [29, 31, 32] and has been used in engineering education contexts [32, 33].

Secondary assessment instruments used for this research include a post-activity qualitative supplement to the CSAS survey [34], a series of post-activity reflective questions, and a set of single-point rubrics developed for the assessment of project proposals, software applications, interactions between clients and student teams [26], peer performance, and written reports. Due to page limit constraints, only the CSAS quantitative evidence will be explored in this paper. Table 1 summarizes the assessment design for the project.



TABLE 1: ASSESSMENT DESIGN FOR THE PROJECT

Artifact	Instrument	L. Obj.	Description
Attitudes Toward Community Service	Community Service Attitude Scale	H	Formative, from program perspective, to examine shifts in the students' service-oriented mindset. First (pre-activity) administration; involves quantitative questions only.
Project Proposal	Project Proposal Rubric	1	Formative. Team proposals are reviewed by education students using single point rubrics during a meeting. Comments are aggregated and promptly returned to the teams.
Preliminary and Revised Application	Software Application Evaluation Rubric	1	Formative and Summative. The programs are evaluated by the clients during a three-week implementation period. Comments are aggregated and promptly returned to the teams. Scores are assigned through collaborative review.
Team Interactions	Peer Evaluation Rubric, Student-Client Interaction Rubric	2	Formative and Summative. Students evaluate one another on their participation in the project and on the quantity and quality of the teaming interactions. Students also provide a self-assessment of their performance. Interaction with the client is assessed using a separate rubric.
Educational Software Application Fair	Software Application Evaluation Rubric	1	Formative. Teams present demos of their applications and deliver a short elevator pitch for judges. Judges include faculty from the education department, teaching assistants for the programming course, and faculty from the college of engineering. Feedback captured using single point rubrics.
Written Report	Written Report Rubric	1, 2	Summative. Used to evaluate the content and mechanics of the report describing the application, <i>but the focus is on the programming methodology</i> . Comments are aggregated and promptly returned to the teams. Scores are assigned through collaborative review.
Attitudes Toward Community Service	Community Service Attitude Scale	H	Formative, from program perspective, to examine shifts in the students' service-oriented mindset. Second (post-activity) administration; includes quantitative and qualitative questions.
Course Feedback	University Student Evaluation Form	H	Formative, from program perspective, to examine shifts in the students' service-oriented mindset through written feedback. Captured at the end of the course.

*Evaluation of Assessment Design:*

To demonstrate the cohesion of the assessment plan, the American Association for Higher Education (AAHE) Principles of Good Practice for Assessing Student Learning [35] is used. The AAHE criteria make up a set of nine “good practices” focusing on not only the gains students exhibit with respect to certain performance criteria, but also how the assessment is positioned in the greater university community, including the stakeholders interested in the results of the evaluation. The following subsections present how each of these principles (shown italicized) is addressed by this project.

1. *The assessment of student learning begins with educational values.* The project focuses on developing student skills in event-driven programming and client-based communications, while also promoting development of a service-oriented entrepreneurial mindset.
2. *Assessment is most effective when it reflects an understanding of learning as multidimensional, integrated, and revealed in performance over time.* Formative assessment across the software application development process is intentionally integrated into the project. Each learning objective is addressed by at least two means of assessment.
3. *Assessment works best when the programs it seeks to improve have clear, explicitly stated purposes.* As the first-year students are still developing programmers, opportunities for formative assessment feedback are placed at critical points in the design process.
4. *Assessment requires attention to outcomes but also and equally to the experiences that lead to those outcomes.* Kolb’s experiential learning model was used to structure the project into spiraled learning opportunities, including multiple face-to-face client interactions and the

use of preliminary individual and team assignments that provide a practice environment for skills development prior to commencing work on the client-based project.

5. *Assessment works best when it is ongoing not episodic.* In addition to the structured formative assessment opportunities built into the experience mentioned in criterion 4, teams are encouraged to contact either their client or the instructor whenever questions arise, which constitutes an unstructured form of formative assessment.
6. *Assessment fosters wider improvement when representatives from across the educational community are involved.* Students are asked beforehand to review all rubrics used in the assessment process as a check for both understanding and clarity. Student feedback for this process has resulted in improvements to the assessment instruments. Design proposals are evaluated via rubrics by a team of education majors and the course instructor to ensure that the feedback considers multiple perspectives. Multiple faculty from the education, engineering, and computer science programs are involved in the formative assessment of the software applications as part of the critical design review process.
7. *Assessment makes a difference when it begins with issues of use and illuminates questions that people really care about.* One of the driving forces behind this research was to show students how their application of learned (primarily computer-based) skills could make a difference in the lives of others. The assessment plan has been designed using a formative approach to provide students with multiple opportunities for feedback, and includes formal touchpoints with clients to maximize the success of the developed software applications.
8. *Assessment is most likely to lead to improvement when it is part of a larger set of conditions that promote change.* This project is part of a larger effort within the College of Engineering to promote the entrepreneurial and service mindset in all its students. As a KEEN (Kern Entrepreneurial Engineering Network) partner, Ohio Northern University is one of several institutions dedicated to transforming undergraduate engineering education by preparing students for success via a combination of attitudes, motivations, and dispositions that provide students the “know-why” that supplements their technical “know-how.” The college also offers multiple EPICS projects for students to participate in as a clear demonstration of a college-wide commitment to service.
9. *Through assessment, educators meet responsibilities to students and to the public.* As feedback is critical for effecting improvements, assessment results are aggregated and returned to the teams in a timely manner. This paper is an example of one responsibility to public, sharing the lessons learned and successes of the project for others to replicate.

## **A Retrospective Analysis of the Project**

This preliminary analysis will make use of the existing data from the CSAS survey to examine the post-activity attitudes and the paired activity deltas across the different offerings of the project. Recall that the hypothesis is that first-year programming students can gain some of the benefits associated with service learning through building client-based relationships with members of various educational learning communities for writing educational software applications, which relates to the stated learning objectives. The independent variable is the year, which relates to how the teams interacted with the clients, while the dependent variables are (1) the post-activity scores, used for validation purposes, and (2) the post-pre activity deltas in the scores, which captures the shift in student mindset toward service. To clarify the differences in the values of the independent variable, a short discussion of the project’s evolution is presented.

*Evolution of the Project in Terms of Client Interaction:*

The cornerstone aspect of this project began in 2014 as a reframing of an existing project in Programming 2. The students were tasked with developing applications as part of a continuing education initiative for high school teachers in the Dominican Republic. Contact with the clients was difficult, so education students served as intermediaries. In the first offering of the project, the communication between the programming course and the education students was done through a single education student who visited the programming teams periodically to provide aggregated feedback and to answer questions. To shift the focus to something more sustainable, in 2015 education students were used as the clients since the university’s ASEE Student Chapter sought to include software applications with their local outreach events but lacked the programming know-how. The single student liaison model was replaced with a more integrated model where a secondary education student was paired with every team such that every team had an education contact. In 2016, an opportunity to partner with a nearby elementary school reframed the client role to teachers located in the same town as the university. Each programming team was still paired with a secondary education student, who served as a liaison in communicating with the teachers since the teachers were mostly unavailable due to their work schedule. Direct communication was limited to written feedback received on the final application submissions. In 2017, the client shifted back to education majors to match the 2015 collaboration model, but with broadened scope to include the early and middle childhood levels.

The three models of collaboration across the four years can be described as liaison (with a non-local client) for 2014, local client for 2015 and 2017, and liaison (with a local client) for 2016. Due to page limit constraints, specific details on each of the iterations are not provided here; however, these details are available from prior publications [18, 24, 36, 37, 38].

*CSAS Survey Results Alignment with the Literature:*

Table 2 presents a summary of reported student attitudes found in the literature, along with the post-activity summaries from this research. In the first column, the numbers presented within parentheses indicate the specific CSAS questions aggregated to form the summary of that attitude. For each study, the number of participants is reported. Due to space limitations, reference numbers are used to identify the CSAS data reported in the literature.

TABLE 2: POST-ACTIVITY CSAS STUDENT ATTITUDES PAIRED WITH THOSE REPORTED IN THE LITERATURE.

7-point Likert Scale	[29]	[30]	[31]	[33]	2014	2015	2016	2017
7 = “strongly agree” <i>n</i> =	332	77	78	17	22	23	21	31
<b>Phase 1: Perceptions</b>								
Awareness (Q1-Q4)	6.21	6.23	5.68	6.27	5.80	6.17	5.95	6.12
Actions (Q5-Q9)	5.61	5.66	5.30	6.07	5.81	6.14	5.82	5.78
Ability (Q10-Q12)	5.42	5.26	5.27	5.95	5.59	5.64	5.71	5.54
Connectedness (Q13-Q18)	5.02	5.10	4.70	5.57	5.11	5.42	5.24	5.09
<b>Phase 2: Moral Obligation</b>								
Norms (Q19-Q23)	6.03	5.92	5.71	6.17	5.71	5.93	5.73	5.88
Empathy (Q24-Q26)	5.61	5.59	5.00	5.75	5.67	5.61	5.51	5.55
<b>Phase 3: Reassessment</b>								
Costs (Q35-Q40)	4.46	4.84	2.78	4.67	5.38	5.35	5.20	5.24
Benefits (Q41-Q46)	5.67	5.81	5.24	6.05	5.57	5.61	5.47	5.56
Seriousness (Q27-Q31)	4.82	4.79	4.47	5.15	5.19	5.17	4.87	4.72
<b>Phase 4: Helping (Q32-Q34)</b>	4.95	4.99	4.61	5.50	5.09	5.28	4.83	4.84

Overall, the post-activity survey data collected with this research (the green-shaded columns in the table) compares favorably with that reported in the literature. Within each attitude save Costs, the research data are contained within and are more tightly clustered than the service learning data reported in the literature. The Costs attitude, while more tightly clustered, falls noticeably above the bounds of the reported CSAS literature data. The Costs attitude here is higher primarily due to the term project format used for this research, which required a greater workload than the assignments reported on in the referenced literature. A more detailed analysis, including the use of paired *t*-tests to identify and discuss individual CSAS questions that demonstrated statistically significant or highly significant changes in student attitudes, can be found in [37].

*CSAS Survey Deltas by Year:*

Figure 3 presents the average deltas for the students engaged with each model of client collaboration. Note the sample sizes are: 20 for liaison with a non-local client, 23 for secondary education clients in 2015, and 29 for the mixed group of clients in 2017. Since the pre-activity CSAS survey was not administered in 2016, this study compares only the deltas across the ten CSAS attitude factors for the different offerings of the project for 2014, 2015, and 2017.

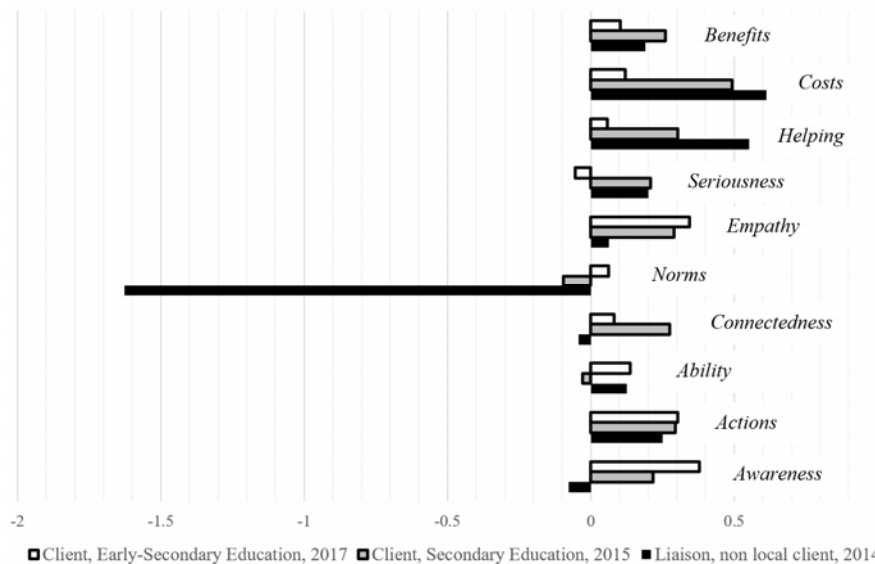


FIGURE 3. CHANGE IN CSAS FACTORS BETWEEN PAIRED PRE- AND POST- TESTS FOR EACH COLLABORATION MODEL

The least successful year in terms of the CSAS factors was the liaison with a non-local client in 2014 which, while gains on individual factors were observed, tended to not produce the highest gain in most factors. The deltas in Helping for the 2014 offering were impressively higher than those for the 2015 and 2017 offerings; however, the differences were not significant. The liaison with a non-local client also produced the most drastic average delta as compared to the other factors; a one way ANOVA was run, and the results revealed highly statistically significant differences between the liaison with a non-local client and the other two years  $\{F(2,69) = 33.26, p < 0.001, \eta^2 = 0.49\}$ . The direct relationship with the client in 2015, conversely, summed to almost entirely positive deltas except for Norms and Ability. Finally, the client model with a broader spectrum of education students produced all positive deltas except for Seriousness. In terms of greatest overall increase across the ten factors, the most successful offering was working directly with the secondary education clients since most of the deltas exceeded the gains (or decreases) with the wider span of educational specialties client liaison model.

### *Limitations:*

Aside from the inability to compare more fully the three models of collaboration, the analysis of the three years is limited in two main areas. First, the comparison assumes the most significant difference between the three years is the relationship between the teams and the client – holding all else equal. Second, the entrepreneurial mindset component was introduced in 2015 to complement the service learning, which is likely a confounding factor. The small sample size is also a threat to the analysis, but the low  $n$  is wholly a function of the class size across the four years of the study. Qualitative evidence to parse out the effects in the model in Figure 3 will be necessary to explain the deltas in more detail – which will be explored in a future publication.

### **Conclusions**

When attempting to adapt a project or idea from one context to another, *transferability* is of interest. The term originates from the evaluation of qualitative inquiry and acknowledges the nature of the paradigm [39]. While results from one study may be logically coherent in the original context, it is advised to refrain from discussing generalizability. Instead, the researcher must make a judgment as to the appropriateness of extrapolating truths from a specific context to his or her own. Adapting instruction and the means of achieving or assessing the attainment of desired outcomes (*e.g.*, through a cornerstone project) requires a similar evaluation. Rather than ending with the results bounded to the context of this work, this section will conclude with a set of salient factors that drove the success and longevity of this effort, for the benefit of any readers who would like to attempt a similar project in their own courses. Student quotes used within this section were obtained from the post-activity qualitative surveys administered after the project.

### *Find the narrative structure of the course and refine as needed to communicate your message:*

A course can be thought of as a serialized performance; for example, telling the “story” of “Data Structures and Algorithms” or “Signals and Systems.” The story is formed from our learning objectives, which are used to determine such things as the cast of characters (topics) and key conflicts (assignments), thereby addressing the *who-*, *what-*, and *where-* type questions. The plot is formed from determining *how* and *when* the key conflicts are set up and resolved. The story and plot combine into a narrative that constructs meaning for the spectators – *i.e.*, the students. Ideally, a story arc is created within the course that leads from exposition, to rising action, to climax, and finally resolution. The implication here is that the course’s narrative structure can influence student attitudes towards one’s profession, and that each component of a course communicates *something*. In the case of the project presented in this paper, this effort originated from an assignment presented near the end of the year-long programming sequence. Students were, perhaps suddenly, expected to realize how their work can make an impact on others, but without any reinforcement through either rising action or resolution activities. Through the research presented here, the course was gradually redesigned to center around a theme of professional responsibility and service by working with clients to achieve a common goal. As one student stated, “this project challenged me to think about someone else rather than myself. I had to think abstractly as to how a sixth grader would think.” By establishing such thematic arcs, opportunities for student engagement are better integrated within a course or a curriculum.

*Select meaningful projects to promote professional identity:*

The word “meaningful” may elicit a collective groan as this suggestion should be obvious, and the definition of the term may be fuzzy. The authors opt to define a meaningful project as an experience that students engage in to (1) satisfy a need (perhaps for a client) and/or (2) advance their professional identity. For this project, the need was manifested in multiple forms throughout the iterations over time with the clients involved; however, the onus was primarily on the instructor to find those with needs that would fit the constructed educational software development story arc. The advancement of a professional identity is subtler and was achieved by immersing students in situations a programmer would experience on the job. In this case, the students were working with clients whose preferences were not necessarily clear. The groups needed to refine their application to address concerns as they arose, which to some extent was dependent on the amount of interaction they had with the client. As one student put it, “you have to keep the client's perspective and goals in mind, and this project helped me to see that.”

*Make use of available resources and opportunities:*

Projects with multiple stakeholders certainly require the assistance of others outside the course. A general suggestion is to simply look around one's institution, especially outside of the department and college, for such opportunities. For this project, collaborating with the Education Department provided a win-win opportunity, as the project's clients are recruited from a Gamification course offered to education majors by one of the authors. Having an identified end user also helps to motivate the programming students, as “it provides a solid incentive to do good work, as someone can actually use what we're doing.” Additionally, many instructors both inside and outside of one's institution are willing to share their resources. In that spirit of sharing, the authors have made all instruments developed for the cornerstone project available for download at the following URL: <https://onueccs.weebly.com/>.

*Engage students in a culture of formative assessments and iterative experiences:*

An intentional design feature of this project was the inclusion of a culture of formative assessment, as facilitated by the nature of Kolb's Experiential Learning Cycle. For instance, structured time was set aside during the regularly scheduled lab session for interactions with clients so teams could receive informal feedback and work side-by-side with their respective clients. Such an approach will be noticed by the students; one of the post-activity comments received was “we had an experience to work on a project that was meant to be used by an actual client rather than just a completion for a grade.” The instruments used to facilitate formative assessment through the process were the single-point rubrics, which incorporated specific and constructive feedback from multiple assessors directly tied to the established criteria. Students were then given time to reflect upon, and then address, the comments received through the conceptualization and experimentation stages of the Cycle.

In closing, the development of the cornerstone project described here has had an overall positive impact, as students appreciated being “given a chance to solve a real world, open ended problem through our coding which will be useful in both our college careers and our careers later in life.” Those interested in implementing a similar project at their institution are welcomed to contact the authors for additional information.

## References

1. D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*. Englewood Cliffs, N.J.: Prentice-Hall, 1984.
2. "EPICS", *EPICS - Purdue University*, 2018. [Online]. Available: <https://engineering.purdue.edu/EPICS>. [Accessed: 25 Jan. 2018].
3. W. C. Oakes, E. J. Coyle, and L. H. Jamieson, "EPICS: A Model Of Service Learning In An Engineering Curriculum," in *Proceedings of the 2000 ASEE Annual Conference & Exposition, 18-21 June 2000, St. Louis, MO* [Online]. Available: ASEE Conferences, <https://peer.asee.org/8361>. [Accessed: 25 Jan. 2018].
4. "Home - Engineers Without Borders USA," *Engineers Without Borders USA*, 2018. [Online]. Available: <https://www.ewb-usa.org/>. [Accessed: 25 Jan. 2018].
5. B. Jaeger and E. LaRochelle, "EWB (2)-Engineers Without Borders: Educationally, a World of Benefits," in *Proceedings of the 2009 ASEE Annual Conference & Exposition, 14-17 June 2009, Austin, TX* [Online]. Available: ASEE Conferences, <https://peer.asee.org/4961>. [Accessed: 25 Jan. 2018].
6. T. J. Kriewall and K. Mekemson, "Instilling the entrepreneurial mindset into engineering undergraduates," *Journal of Engineering Entrepreneurship*, vol. 1, no. 1, pp. 5-19, 2010.
7. KEEN, "KEEN - Mindset Matters", *Engineeringunleashed.com*, 2018. [Online]. Available: <https://engineeringunleashed.com/Mindset-Matters.aspx>. [Accessed: 25 Jan. 2018].
8. P. Singh and M. V. Moncada, "Instilling the Entrepreneurial Mindset by International Development Project Work," in *Proceedings of the 2015 ASEE Annual Conference & Exposition, 14-17 June 2015, Seattle, WA* [Online]. Available: ASEE Conferences, <https://peer.asee.org/24310>. [Accessed: 25 Jan. 2018].
9. B. Rosmaita, "Making service learning accessible to computer scientists," *ACM SIGCSE Bulletin*, vol. 39, no. 1, pp. 541-545, 2007.
10. P. Sanderson and K. Vollmar, "A primer for applying service learning to computer science," *ACM SIGCSE Bulletin*, vol. 32, no. 1, pp. 222-226, 2000.
11. C. Traynor and M. McKenna, "Service learning models connecting computer science to the community," *ACM SIGCSE Bulletin*, vol. 35, no. 4, pp. 43-46, 2003.
12. L. Jamieson, "Service learning in computer science and engineering," *ACM SIGCSE Bulletin*, vol. 34, no. 1, pp. 133-134, 2002.
13. J. Tan and J. Phillips, "Incorporating service learning into computer science courses," *Journal of Computing Sciences in Colleges*, vol. 20, no. 4, pp. 57-62, 2005.
14. R. Ferguson, C. Liu, M. Last and J. Mertz, "Service-learning projects: opportunities and challenges," *ACM SIGCSE Bulletin*, vol. 38, no. 1, pp. 127-128, 2006.
15. M. A. L. Egan and M. Johnson, "Service learning in introductory computer science," in *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education - ITiCSE 10, 26-30 June 2010, Ankara, Turkey*, pp. 8-12, 2010.
16. J. B. Adams and E. Runkles. "'May we have class outside?'" Implementing Service Learning in a CS1 Curriculum," *Journal of Computing Sciences in Colleges*, vol. 19, no. 5, pp. 25-34, 2004.
17. Y. Kortsarts and J. Rufinus, "Integrating a Project Component into the Introductory Computer Science and Information Systems Curriculum," *Information Systems Education Journal*, vol. 5, no. 6, pp. 1-11, 2007. [Online]. Available: <http://isedj.org/5/6/>. [Accessed: 25 Jan. 2018].
18. J. K. Estell, D. Reeping, and H. Sapp, "Curiosity, Connection, Creating Value: Improving Service Learning by Applying the Entrepreneurial Mindset," in *Proceedings of the 2016 ASEE Annual Conference & Exposition, 26-29 June 2016, New Orleans, LA* [Online]. Available: ASEE Conferences, <https://peer.asee.org/26621>. [Accessed: 25 Jan. 2018].
19. R. Greenaway, "The Active Reviewing Cycle | Reviewing Skills Tutorial," *Reviewing.co.uk*, 2018. [Online]. Available: <http://reviewing.co.uk/learning-cycle/the-active-reviewing-cycle.htm>. [Accessed: 25- Jan- 2018].
20. "Common Core State Standards Initiative," *Corestandards.org*, 2018. [Online]. Available: <http://www.corestandards.org/>. [Accessed: 26 Jan. 2018].
21. "Next Generation Science Standards," *Nextgenscience.org*, 2018. [Online]. Available: <https://www.nextgenscience.org/>. [Accessed: 26 Jan. 2018].
22. J. Moule, *Killer UX Design*. Collingwood, Vic.: SitePoint, 2012.
23. J. Fluckiger, "Single Point Rubric: a Tool for Responsible Student Self-Assessment," *Delta Kappa Gamma Bulletin*, vol. 76, no. 4, pp. 18-25, 2010.

24. J. K. Estell, H. Sapp, and D. Reeping, "Work in Progress: Developing Single Point Rubrics for Formative Assessment," in *Proceedings of the 2016 ASEE Annual Conference & Exposition, 26-29 June 2016, New Orleans, LA* [Online]. Available: ASEE Conferences, <https://peer.asee.org/27221>. [Accessed: 25 Jan. 2018].
25. E. Ries, *The Lean Startup*. New York: Crown Business, 2011.
26. J. K. Estell and S. Howe, "Development and Use of a Client Interaction Rubric for Formative Assessment," in *Proceedings of the 2017 ASEE Annual Conference & Exposition, 25-28 June 2017, Columbus, OH* [Online]. Available: ASEE Conferences, <https://peer.asee.org/28157>. [Accessed: 25 Jan. 2018].
27. D. Fisher and N. Frey. *Checking for Understanding: Formative Assessment Techniques for your Classroom*. Alexandria, VA: ASCD, 2015.
28. A. Leskes, "Beyond confusion: An assessment glossary", *Peer Review*, vol. 4, no. 2/3, pp. 42-43, 2002.
29. A. Shiarella, A. McCarthy and M. Tucker, "Development and Construct Validity of Scores on the Community Service Attitudes Scale," *Educational and Psychological Measurement*, vol. 60, no. 2, pp. 286-300, 2000.
30. C. A. Downey, "Student Research in an Introductory Psychology Course: Outcomes of Two Experiential Learning Projects and Implications for Instruction of Human Subjects Research," *The Journal of Effective Teaching*, vol. 13, no. 2, pp. 21-37, 2013.
31. E. Bauer, B. Moskal, J. Gosink, J. Lucena and D. Muñoz, "Faculty and Student Attitudes toward Community Service: A Comparative Analysis," *Journal of Engineering Education*, vol. 96, no. 2, pp. 129-140, 2007.
32. B. Perry, R. Osbaldiston and J. Henning, "Tests of the Validity and Reliability of the Community Service Attitudes Scale," *Journal of College Student Development*, vol. 55, no. 7, pp. 726-731, 2014.
33. A. Bielefeldt, B. Amadei, and R. Sandekian, "Community Service Learning Attitudes of Engineering Students Engaged in Service Learning Projects," in *Proceedings of the 2008 ASEE Annual Conference & Exposition, 22-25 June 2008, Pittsburgh, PA* [Online]. Available: ASEE Conferences, <https://peer.asee.org/4319>. [Accessed: 25 Jan. 2018].
34. "Service-Learning Student Post-Assessment," *Towson.edu*, 2018. [Online]. Available: <https://towson.edu/provost/initiatives/leadership/teaching/servicelearning/faculty/documents/survey.pdf>. [Accessed: 29 Jan. 2018].
35. A. Astin, T.W. Banta, K.P. Cross, E. El-Khawas, P.T. Ewell, P. Hutchings, and E. Moran, "Principles of Good Practice for Assessing Student Learning," *AAHE Bulletin*, vol. 45, no. 4, 1992.
36. D. Reeping, K. Reid, and J. K. Estell, "Work in Progress: Providing Continuing Education for Teachers in the Dominican Republic Using Online Modules Developed through a First-Year Capstone Project," in *Proceedings of the 2014 ASEE Annual Conference & Exposition, 15-18 June 2014, Indianapolis, IN* [Online]. Available: ASEE Conferences, <https://peer.asee.org/23325>. [Accessed: 25 Jan. 2018].
37. J. K. Estell and D. Reeping, "Providing Authentic Experiences in the First Year: Designing Educational Software in Support of Service Learning Activities," in *Proceedings of the 2015 ASEE Annual Conference & Exposition, 14-17 June 2015, Seattle, WA* [Online]. Available: ASEE Conferences, <https://peer.asee.org/24617>. [Accessed: 25 Jan. 2018].
38. J. K. Estell, D. Reeping, H. Sapp, and A. Seda, "Work in Progress - First-Year Programmers Developing Software Applications for a STEM Academy Workshop: An Exercise in Collaborative Service Learning," in *Proceedings of the 7th Annual First Year Engineering Experience Conference, 2-4 Aug. 2015, Roanoke, VA* [Online]. Available: ASEE Conferences, <http://fyee.asee.org/FYEE2015/papers/5005.pdf>. [Accessed: 25 Jan. 2018].
39. J. W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Thousand Oaks, CA: Sage Publications, 2014.