2005

EPICS: A Service Learning Program at Butler University

Jonathan P. Sorenson
Butler University, jsorenso@butler.edu

Panos K. Linos
Butler University, linos@butler.edu

EPICS: A Service Learning Program at Butler University

Jonathan P. Sorenson
Computer Science and Software Engineering
Butler University
Indianapolis, Indiana 46208
sorenson@butler.edu
http://www.butler.edu/~sorenson

Panagiotis K. Linos
Computer Science and Software Engineering
Butler University
Indianapolis, Indiana 46208
linos@butler.edu
http://blue.butler.edu/~linos

***Abstract*— In this paper we present our experiences teaching EPICS (Engineering Projects In Community Service) at Butler University, a small, private university, from within the Department of Computer Science and Software Engineering.**

The EPICS program began at Purdue University in 1995. The idea behind EPICS is to have undergraduate students earn college credit for working on long-term, multi-semester projects to benefit charity and non-profit organizations. The projects are student-driven, under faculty supervision.

There are many good reasons for having an EPICS program in an undergraduate computer science major. It is excellent for leveraging knowledge from other areas of computer science such as databases, networks, operating systems, and of course software engineering. The students are highly motivated because the project is real: there are real clients who use the software, making the software lifecycle come to life. Students practice teamwork, project management, professionalism, and communication skills. In our paper, we share feedback from our students on what EPICS means to them. At Butler, EPICS has been a success.

Our EPICS program started in the Fall 2001 semester. We now have two ongoing projects: Spanish-In-Action (SIA), with Spanish middle school teachers from Crispus Attuchs Middle School in Indianapolis as clients, and Social Assets and Vulnerabilities Indicators (SAVI), with the POLIS Center at IUPUI as the client. We describe both projects in some detail in our paper.

EPICS currently counts towards both the computer science major and the software engineering major as an elective at Butler. Our department has about 50 students and 4 full-time faculty, and each semester we have roughly 15 students enrolled in EPICS. We elaborate on how EPICS fits into our curriculum and provide details on how we deliver this course in our paper.

Index Terms — Service Learning, EPICS, Computer Science Education, Software Engineering Education

I. Introduction

We were motivated to begin a service-learning program in our department after interacting with representatives from Purdue's very successful program, EPICS (Engineering Projects for Community Service) [1], [2], [3] and Butler's own Center for Citizenship and Community [6]. Service Learning can bring a higher level of meaning to student coursework, and working with the community brings rewards as well.

In this paper, we describe our experiences implementing our own version of EPICS in computer science and software engineering.

II. What Is EPICS?

The features of an EPICS program are as follows:

- *Long Term Projects.* EPICS projects span multiple semesters; the project itself may continue for years, and students are encouraged to participate for multiple semesters.
- *Vertically Integrated Teams.* In a particular semester, a student team working on a project normally includes all levels of students, from freshmen to seniors.
- *Multidisciplinary Teams.* Students need not be computer science or software engineering majors to participate, although students majoring or minoring in these areas do dominate the teams.
- *Charity/Non-profit Clients.* The clients for EPICS projects are members of charity or not-for-profit organizations.

EPICS is a course the students take – they register and receive a grade. The fact that students can take the course multiple times provides for project continuity and a sense of living history.

Below we explain why we feel this model for service learning is a good fit for computer science, and we also explain how we deliver this course.

There are, of course, many other ways to provide a service-learning or community service experience for computer science students. For some examples, see [7], [8], [9], [10], [11], [12], [13].

III. Why Have an EPICS Program?

We have discovered that there are many good reasons for implementing the EPICS model for service learning in computer science. We feel it is excellent pedagogy. Yet, in the EPICS course, no specific computer science concepts or knowledge is taught to the students. What, then, makes this course so worthwhile that we've written this paper about it? As we elaborate on this, we quote freely from our EPICS course evaluations.

- *Real Projects.* EPICS projects have real, outside clients with real needs to fill, and the software that is developed will be used. This level of reality is not typical for coursework; in most courses the programming projects

are done to specifications set by the instructor, and they are "thrown away" at the end of the semester. EPICS projects provide experiential learning; they allow the student to apply what she knows in a realistic but protected environment.

> "This course is great; it provided a real world experience."
> "This course is very important and useful for CSSE students. I believe this course is designed for students to prepare them for real life projects, and project management. This course must be highly recommended in CSSE major, because it gives students a taste or a piece of the real world work environment."

- *Motivation.* Educators know that motivated students learn much, much more that unmotivated students. Our students know that the work they put into the project will help the community; this gives them motivation and helps make the project fun.

  > "The course was good. I feel like I learned more under my own motivation than I learned in other classes, specifically because I was motivated."
  > "EPICS continues to be an enjoyable class as well as a good learning class."

- *Technical Development.* Although it is true that the course goals for EPICS do not include teaching any specific computer science content or concepts, students do practice their technical knowledge and, in many cases, teach themselves specific technologies (such as software or languages) that they had not encountered before. We particularly like the fact that some computer science content is reinforced through EPICS; for our projects, the areas include databases, operating systems and networks, programming languages, and of course software engineering. Students taking EPICS concurrently with a database course, for example, end up learning more in both courses as they play off each other.

  > "I learnt even more about databases this semester. I think it was a good semester."
  > "I have had a lot of fun in EPICS this semester. Not only did I learn a lot about technical aspects (such as coding in Flash, UML, etc.) but I learned a lot about working in a team environment. Although I put forth a lot of effort for the class, I didn't feel as if I didn't get anything out of it."

- *Soft Skills.* Many of the soft skills, such as teamwork, project management, professionalism, and communication skills, are exercised in EPICS. The students run the projects themselves (as we'll see below) and they are the ones who communicate with the client. These skills are difficult to teach in a regular course driven by content.

  > "This course is invaluable in gaining project experience and working in a team environment."
  > "...Ultimately, this leads to the students running their own class, just like a small business. Of all the courses I've taken, EPICS offers the most real world experience. This makes the class a large asset in college preparing students for what a job will be like in the real world."

  One of our recent graduates, who had participated in EPICS for four semesters, returned to have lunch with our current students and to share her experiences. She praised EPICS, and pointed out that her experiences working with teams in EPICS provided excellent material for discussion during job interviews.

- *Low Risk.* In the end, our clients understand that they are working with students. They know that the effort is there, but the results will not always be perfect (perhaps this is not so different from a real company). If the students are not able to meet their semester goals, or if there is a major bug in the software, we save that as a goal for the following semester and the project marches on. Unlike in the "real world," we don't lose the contract or have to take pay cuts or lay anyone off. Our clients are, for the most part, happy with what they get.

  Here are some quotes from our client evaluation forms:

  > "Thank you so much for all your work this semester."
  > "Thanks!!!"
  > "Excellent – very responsive to needs we have."
  > "We are more than happy to work with you all."

- *Attracting Underrepresented Groups.* Service learning programs like EPICS tend to attract female and minority students. This has definitely been our experience. For example, our very first EPICS team in the Fall 2001 semester was composed of 5 women and 1 African-American student. Excluding freshmen, all of our female students have participated in at least one semester of EPICS, and most of our team leaders have been women and/or from ethnic minorities.

IV. EPICS IN THE CURRICULUM

As mentioned above, we offer EPICS as a regular course in our department. In fact, we offer 6 courses in EPICS: two at each of the 200, 300, and 400 levels, including one for 2 credit hours and one for 3 credit hours. These courses carry the SL university-wide rubric for service-learning courses. The courses may be repeated for credit.

We currently accept the 300 and 400-level versions of EPICS as electives for both our computer science and software engineering majors. Each level of EPICS requires taking the previous level. We normally expect students registering at the 400-level to take team leadership roles, whereas 200-level students are expected not to know much. The only prerequisite for 200-level EPICS is one semester of programming. The 300 and 400-level courses require more CSSE courses, including some software engineering.

We offer 2-credit and 3-credit versions of the courses in an attempt to accomodate our students' schedules. We are mindful

Fig. 1. Quick Drop game screenshot.

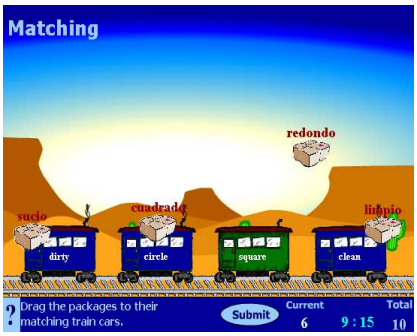

Fig. 2. Quiz System screenshot.

that EPICS is elective, and do what we can to encourage students to register. We normally do not grade the 2 and 3-credit versions any differently.

Note that students who are currently taking EPICS make brief visits to other CSSE courses to encourage students to register for EPICS the following semester. Having student recruiters is very effective, especially with the introductory programming courses.

For specifics on our majors and courses, we invite the reader to visit our departmental web pages [14].

V. Current EPICS Projects

We have two active EPICS projects at Butler [15], [16]: *Spanish-In-Action* and *SAVI*.

A. Spanish-In-Action (SIA) and Quick Drop

The SIA project maintains and develops a web-based computer game, called *Quick Drop*, to help students at Crispus Attucks Middle School (CAMS) [17] learn Spanish vocabulary. Students at CAMS are given iBooks to help with their studies, and this project was intended to make good use of these machines. The SIA project consists of the following pieces:

- The *Quick Drop* game, which is written in Macromedia Flash and ActionScript. (See Figure 1.)
- The Web Administration System (WAS), which allows teachers to use the game by providing vocabulary and tracking student performance; it is written in PHP, HTML, and uses a MySQL relational database.
- The Quiz System, which provides a way for teachers to give quizes using an environment similar to that of the game; it uses Flash, ActionScript, PHP, HTML, and the database. (See Figure 2.)
- The Adventure Game, which is in the very early stages of design, and intends to provide a way for students to interact with Spanish-speaking characters in a Spanish-based cultural setting.

The clients are the Spanish teachers at CAMS.

This project was the first EPICS project at Butler, and was started in the Fall 2001 semester. In the Spring 2002 semester, a prototype of *Quick Drop* won first place at the Indiana Student Software Awards Competition. In the Spring 2004 semester, *Quick Drop* won the *Best non-Purdue EPICS Project* award at the EPICS Idea-to-Product competition, sponsored by Purdue's Burton D. Morgan Center for Entrepreneurship [18].

Future work on SIA includes ironing out bugs in the web support and quiz systems, releasing a new and updated version of *Quick Drop*, completing the design of the adventure game, and porting the systems to other languages like French and Greek.

B. SAVI Query Builder

The SAVI project is involved in developing a site-specific intelligent search engine. Our client is the group at the POLIS center that supports SAVI. SAVI-Interactive [19] is a dynamic community information system that seeks to enhance capacity and improve decision-making in Central Indiana communities and especially in organizations and agencies that serve human needs [20], [21]. SAVI (Social Assets and Vulnerabilities Indicators) provides information from approximately 40 data providers. Also, it entails tools to access and analyze such information, and it offers user support and training. It is hosted by the POLIS Center of Indianapolis [22], which is an academic research center with a practical and applied orientation [23], [24]. The center concentrates on issues related to metropolitan Indianapolis and other mid-sized American cities. Its mission entails efforts to build the capacity of communities to develop knowledge about them, to build collaborations, and to create and apply information innovatively and effectively.

Some of the specific goals of the SAVI project at the POLIS Center include:

- Build community capacity by empowering citizens and organizations with data and training them on its use and helping them use for more effective decision-making.
- Build community capacity to make positive changes through more effective policies, programs, and actions.
- Improve decision making by providing relevant data and tools.
- Create a community resource for information by developing SAVI as a community information system that provides:

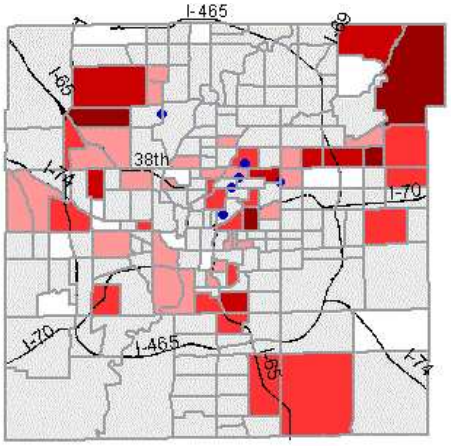

Fig. 3. The SAVI tools plot statistical data on Indianapolis maps.

- Relevant information.
- Tools to access and analyze information easily.
- Training and user support on how to use these resources effectively.
- Outreach and education to increase awareness of its resources.
- Integration of a variety of specialized resources across the Indianapolis metropolitan area.

The EPICS-SAVI Query Builder project was created in the spring of 2002 at Butler University as an effort to collaborate with the SAVI-interactive project at the POLIS center. Since then, our EPICS team has designed and implemented various ”Google like” database search features. The SAVI Query Builder project is focusing on the development of algorithms, standards, tools and effective search methods that can facilitate a variety of database queries. The SAVI Query-Builder has been implemented using the .NET framework. It is designed in an object-oriented fashion consisting of several classes. These classes implement various important features, and enhance the capability to conduct an efficient database query. The first feature of the system supports noise removal and unwanted characters by filtering the input character string in order to provide a more effective search string. Using the fine-tuned input string, the next feature creates a Boolean expression based upon typical logical operators such as AND, OR, and NOT. In addition, any text included in quotes is left unchanged and it is treated as a single unit by the Boolean operators. The last part of the Query Builder’s functionality supports the conversion of the Boolean expression to an Oracle database query, and can be extended easily to work with other types of databases. Finally, the Query Builder makes use of an external XML file, which contains all of the ”noise” characters and/or keywords to be removed from the input string. The information found in this file is also stored in an access database. So, should the customer decide to add additional ”words to be removed” from the search query, he/she can add them to the database file and regenerate the XML file.

VI. Delivering the EPICS Course

In this section we briefly describe how we teach the EPICS courses, but before we go into detail, we briefly describe our department.

The CSSE department at Butler University has 4 full-time faculty and about 50 majors total. We have no graduate program. Butler University is a comprehensive, undergraduate university with about 3600 students with four professional colleges and the College of Liberal Arts and Sciences (LAS). The CSSE department is in LAS. In a given semester, EPICS typically has about 15 students enrolled: 10 or so with SIA, and 5 or so in SAVI.

Each EPICS project gets its own section of the EPICS course, so students choose the project they wish to work on when they register. If a student changes his mind, he can drop/add into the correct section during the first week of the semester.

We have normally identified the student (usually a senior) who will serve as project leader before the semester starts, and the first order of business is to form the students into teams, with team leaders. For example, on the SIA project, we might put 3 students into a Quick Drop upgrade team, 4 students on the Web Administration team, and so forth. Each team will have a student leader. SAVI normally functions as a single team.

The teams spend about a week deciding on goals for the semester, which must be approved by the instructor. Then they get to work. The students run the course. Instructors normally only serve as mentors and advisors; there is usually no lecture or any other imposed structure on class time.

> “I really enjoy this course and like the structure that has evolved.”
>
> “There weren’t any exams.”

To evaluate the students (that is, assign grades) we do the following:

1) Students fill out a peer-evaluation form near the end of the semester (with a practice run midway through the semester). They evaluate other team members plus the project leader.
2) The instructor interviews each of the students for 15-30 minutes to share the results of the peer evaluations and to discuss any issues that may arise. (Practice interviews are done midway through the semester.)
3) The instructor grades team deliverables at the end of the semester. What the deliverables are depends on the team goals, but can include source code, requirements or design documents, websites, and so forth.
4) Each project group will make two presentations near the end of the semester: one *client presentation* for the clients, explaining what was accomplished during the semester and where the project is going, and one more informal *technical presentation* for the instructor and other interested faculty and students giving technical details. If the client is technically knowledgeable, then the two presentations can be combined.

5) Clients fill out a project-specific evaluation form.
6) Each team receives a grade based on their deliverables, presentations, and client evaluations.
7) Each student receives a grade relative to the team grade. If their peer evaluations rate them above average for the team, then the team grade would be a minimum. The idea is for the average grade of the students on a team to match the team grade. If peer evaluation appears to be inaccurate, the instructor can fill out a peer evaluation form for a particular student, give it an appropriate weight, and include that in the calculation. This must only be done in unusual circumstances and only in a transparent way to maintain fairness.

We feel this way of arriving at grades promotes teamwork, leaves final control of the course in the instructor's hands, yet gives the students the freedom to run the course the way they like.

> "This course is designed to be completely student run. Leaders are elected and groups are formed with little to no input from the professor. He is only here for supervision or for if an emergency should arise. However, his non-interference is also his strong point as it forces students to take matters into their own hands, rather than leading them through it like a child. Sorenson still controls the final grade, but this is influenced by peer evaluations, putting even more power in the hands of the students. Ultimately, this leads to the students running their own class, just like a small business. . . ."

This does not mean that the instructor does not work. Teaching EPICS can be very time-consuming, especially in the first one or two semesters a project is going, and at the end of the semester when all that grading has to be done. It is also imperative that the instructor keep close tabs on what is happening in each of the groups. Often problems, such as acquiring needed software, or working with the campus computer center, require faculty members to solve. The clients also need to feel that the faculty instructor is watching and monitoring what the students are doing.

One possible teaching load model is to credit an instructor with teaching a regular course for EPICS during the first year of a project. Once a group of returning, knowledgeable students is established, and the patterns of how to run things are established, the course might be treated as a laboratory course after that, perhaps counting as one or two credit hours.

VII. Conclusion

We have found EPICS to be a fun, rewarding, and highly worthwhile way to provide our students with a pedagogically sound service learning experience in computer science. We hope this paper encourages you to try something similar in your program.

Acknowledgment

The authors would like to thank our students, for their dedication, diligence, and hard work; our clients and partners at CAMS and the POLIS Center for their willingness to work with us; the folks from EPICS at Purdue University, such as Ed Coyle, Bill Oakes, Leah Jamieson, and Heather Connell, for their continuing support [2]; the Center for Citizenship and Community at Butler [6], and the Sallie Mae Foundation for their generous support.

References

[1] E. J. Coyle, L. H. Jamieson, and W. C. Oakes, "EPICS: Engineering projects in community service," *International Journal of Engineering Education*, vol. 21, no. 1, pp. 139–150, Feb. 2005.
[2] "EPICS at Purdue University website," 2005, http://epics.ecn.purdue.edu.
[3] "EPICS national website," 2005, http://epicsnational.ecn.purdue.edu.
[4] J. Reid and E. Slazinski, "Successful knowledge transfer and project deployment in a service learning program," in *CITC4 '03: Proceeding of the 4th conference on Information technology curriculum*. ACM Press, 2003, pp. 222–225.
[5] L. H. Jamieson, "Service learning in computer science and engineering," in *SIGCSE '02: Proceedings of the 33rd SIGCSE technical symposium on Computer science education*. ACM Press, 2002, pp. 133–134.
[6] M. Brabant and D. Braid, "The Center for Citizenship and Community of Butler University," 2003, brochure. http://www.butler.edu/centerforcc.
[7] J. L. Gersting and F. H. Young, "Service learning via the computer science club," *SIGCSE Bull.*, vol. 30, no. 4, pp. 25–26, 1998.
[8] P. Sanderson and K. Vollmar, "A primer for applying service learning to computer science," in *SIGCSE '00: Proceedings of the thirty-first SIGCSE technical symposium on Computer science education*. ACM Press, 2000, pp. 222–226.
[9] J. Skon, "Computing in Papua New Guinea: lessons learned about computer science service learning projects," *J. Comput. Small Coll.*, vol. 17, no. 1, pp. 31–38, 2001.
[10] C. Schahczenski, "Computer science, nonprofits and service learning," in *32nd ASEE/IEEE Frontiers in Education*. IEEE, November 2002, p. 125.
[11] P. Sanderson, "Where's (the) computer science in service-learning?" *J. Comput. Small Coll.*, vol. 19, no. 1, pp. 83–89, 2003.
[12] C. Traynor and M. McKenna, "Service learning models connecting computer science to the community," *SIGCSE Bull.*, vol. 35, no. 4, pp. 43–46, 2003.
[13] J. Tan and J. Phillips, "Incorporating service learning into computer science courses," *J. Comput. Small Coll.*, vol. 20, no. 4, pp. 57–62, 2005.
[14] "Butler University Department of Computer Science and Software Engineering website," 2005, http://www.butler.edu/csse.
[15] P. K. Linos, S. Herman, and J. Lally, "A service-learning program for computer science and software engineering," in *ITiCSE '03: Proceedings of the 8th annual conference on Innovation and technology in computer science education*. ACM Press, 2003, pp. 30–34.
[16] "EPICS website (SIA and SAVI)," 2005, http://epics.butler.edu.
[17] "Crispus Attucks Middle School website," 2005, http://www.518.ips.k12.in.us.
[18] "EPICS I2P competition website," 2005, http://ims.ecn.purdue.edu/entrepreneurship/I2PCompetition.php.
[19] "SAVI Interactive Website," 2005, http://www.savi.org.
[20] S. Foster, "The SAVI community connections project," *Digital Divide Network*, Apr. 2001, http://www.digitaldivide.net/articles/view.php?ArticleID=337.
[21] S. Kandris and K. E. Frederickson, "The development of the social assets and vulnerabilities indicators (SAVI) database," in *Proceeding of the 2001 ESRI User Conference*, 2001, http://gis.esri.com/library/userconf/proc01/professional/papers/pap846/%p846.htm.
[22] "The POLIS Center," 2005, http://www.polis.iupui.edu/tpc.
[23] M. Riner, C. Cunningham, and A. Johnson, "Public health education and practice using geographic information system technology," *Public Health Nursing*, vol. 21, no. 1, pp. 57–65, 2004.
[24] R. Vernon, "Web-based support for program planning and research," *Journal of Technology in Human Services*, vol. 22, no. 2, pp. 81–87, 2003.