Product-Based Learning in Software Engineering Education

Eric D. Ragan, Stephen Frezza, and Jeremy Cannell
ericragan@vt.edu, frezza001@gannon.edu, cannell001@gannon.edu

Abstract - **Problem-based learning is a pedagogy that employs open-ended problems as a means of increasing student motivation and interaction in addition to achieving the intended educational goals. We present product-based learning as an extension of collaborative problem-based learning by focusing on the design and development of a product as a solution to a proposed problem. In this approach, the instructor becomes an active member of the project team, working with the students to achieve the product vision. Through this method, students take control of their own learning while working together towards group success. Such a course design allows students to become active learners in achieving learning objectives, while at the same time providing project management and team-working experience. We explain the concept of product-based learning and demonstrate the process through the design and development of an educational software application. This project shows that the product-based method can promote positive attitudes, social interaction, and intentional learning in students while motivating them to achieve the intended learning objectives.**

Index Terms - Problem-based learning, collaborative learning, computing engineering education.

INTRODUCTION

Focusing on real-world, open-ended problems, problem-based learning is a pedagogy that has been shown to increase student motivation and interaction [1]. This method requires cooperation among students and promotes higher level thinking that is not always necessary when learning by more traditional, independent methods [6, 13]. Furthermore, the collaborative component provides the advantages of supporting the development of social and team-working skills [8]. This high level of student engagement, interest, and motivation promotes intentional learning [3] in students, leading them to become purposeful and self-directed in their pursuit of learning as a goal [4, 10]. Purpose implies that students have clear goals, an understanding of the processes they are undertaking, and that they take appropriate actions to achieve their goals. Further, purpose implies intention in the students' actions. Becoming such an intentional learner means developing self-awareness about the reason for study and the learning process itself. Intentional learners are integrative thinkers who can see connections in seemingly disparate information and draw on a wide range of knowledge to make decisions [3, 4, 9].

Striving to encourage such attitudes, educators have demonstrated that problem-based learning can motivate better than more traditional, less interactive pedagogies and indeed promote greater material comprehension for computer programming courses [e.g., 1, 2, 5]. Despite its strengths, we feel that the problem-based approach can be further refined for technological and engineering sciences. As an extension to problem-based learning, we introduce product-based learning, which requires the development of a real-world product as a solution to an open-ended problem. This approach provides further motivation by establishing the goal of satisfying a real user base. Not only do students become responsible for their own learning and the success of the team, but they must also take responsibility for a quality product to be used by known end users.

Additionally, product-based learning employs mutual development between teacher and students. In other interpretations of project-based learning, the instructor generally only serves as a guide to help lead the students to achieving their own learning [13]. By contrast, our method integrates the instructor as part of the team, allowing a mutual development process. This allows the instructor to become more aware of student thinking, allowing a close connection to the learning and development processes. Not only do the students benefit from more helpful and relevant guidance, but the instructor benefits by experiencing the same interaction and motivation that drives the students.

To help illustrate how the product-based learning approach can be an effective method for teaching mid-level technological and engineering topics, we present the approach as applied to a computer science course in object-oriented programming using Java. Because student learning varies with motivation and self-learning [10], this approach began with a motivated project and the stated expectation of the development of real-world deliverables. In addition to deeply engaging students and motivating intentional learning, the product-based approach allowed for a smooth integration of a variety of topics under a single project. The approach can serve as an alternative pedagogical method for classes that are traditionally taught through the lecturing instructor-leader model. While collaborative, project-based approaches are commonly employed in engineering capstone courses, we advocate their use in mid or low-level courses which have specific, fundamental learning objectives even beyond those of project management or product design and construction.

BACKGROUND: COLLABORATIVE LEARNING

Many educators have investigated the benefits of collaborative learning exercises as compared to more traditional independent or lecture-based methods [*e.g.,* 2, 6, 8, 11, 12]. In addition to promoting a deeper level of learning of the primary material, collaborative learning can help students to develop healthy social relationships and learn important team working skills [8]. Educators have even been able to provide empirical evidence that students who learn through cooperative activities can do just as well or better on traditional tests as students taught through non-collaborative methods [e.g., 2, 6, 12]. Both Beck [2] and Sabin [12] provide evidence that the cooperative learning approach leads to significantly better learning results in introductory computer science courses. Furthermore, comparisons of student survey responses for similar experiments suggest that the collaborative learning experiences were preferred by the students [6, 12].

Problem-based learning is one teaching approach that often makes heavy use of collaborative learning [1]. This approach typically involves students working and learning in small groups with teachers serving as facilitators [13]. While more traditional teaching methodology often relies on lectures and small exercises, problem-based learning focuses on large, open-ended problems [1]. These problems, like many real-world problems, lack clearly defined criteria and absolute right or wrong answers [5]. In problem-based learning, it is the process of problem solving that drives the learning [13]. Barg *et al.* [1] demonstrated that problem-based learning could be successfully used to teach introductory programming, finding evidence that students developed better understandings of basic concepts and better attitudes about the material. Daniels [5] analyzed a collection of several successful documented open-ended group projects in areas related to systems and engineering.

One of the biggest advantages of these types of educational approaches is that the high level of engagement increases the levels of student motivation and interest [5, 13]. Because students must learn the material in order to solve the problem, they must actively strive to learn. This concept, in which learning itself becomes a goal, is known as intentional learning [3, 10]. Activities involving high interaction and cognitive processing promote this attitude [7, 10]. Additionally, many students may benefit from the planning and management necessary to attain their educational goals along the way, teaching them how to learn [4, 10].

Our product-based approach varies the problem-based approach by integrating the instructor as an equal partner into the problem-based method. In this development of the product, the instructor acts as part of the team. Part of the distinction is that the instructor deliberately does not act as the team leader, but more as an equal participant and experienced guide. This distinction in the learning environment is subtle but important, and requires students to take ownership of the project and for faculty to relinquish the appearance of control over the final product. The students on the project are directed to develop the product vision, and (with guidance) manage those aspects of the product development related to achieving the course objectives. Another distinction from problem-based approaches is the promise of, and interaction with product stakeholders.

PRODUCT-BASED LEARNING: COURSE SELECTION

While product-driven development projects are common components of a capstone or software engineering courses in which the project management and production of deliverables are the primary goals, our method enables a similar approach to be adapted to low and mid-level courses as a means of teaching separate course objectives. The only requirement for course selection is that the course objectives include learning tools or techniques that can be applied product development, as is common with many engineering courses.

Our example is taken from an undergraduate object-oriented programming course. The course (CIS286) was aimed to teach Java OOP to sophomore and junior computer-science and software engineering students in a small classroom setting (Figure 1). This course was selected for applying product-based learning because the development process for a large application could naturally explicate the benefits of the advanced object-oriented design principles on which the course's learning objectives are based. Additionally, it was believed that the students, sophomores and juniors with some programming expertise, could benefit from the experience attained through engineering a software application significantly larger than those of typical class assignments. Moreover, the open-ended nature of such an approach held potential for the unification of a broad variety of complimentary software development principles through a single project.

The product-based approach involves dividing the class into small groups to design and develop individual design components. Because of the small size of the class (9 students), these "groups" were small, usually consisting of only one or two students. We argue that larger sections can use a similar method using larger groups or more components for more complex products. As a result, the method could prove to be even more beneficial in larger classes than in small classes, as it requires additional collaboration and coordination within groups as well as between groups.

Successful implementation requires an instructor who can be adept in guiding system architecture development so that student teams are correctly aligned with their product components. The process will require balancing the number of components, system complexity, and the number of students so that the class and groups can function coherently while allowing optimization of individual learning. We recognize that this may not be possible for exceptionally large classes, as management and coordination challenges could become too unwieldy to stimulate individual learning.

PRODUCT-BASED LEARNING: SELECTING A PROBLEM

Key to the use of this approach is the selection of a suitable problem for which students have sufficient experience to actively participate in the creation and refinement of the product vision. Ideally, the problem should be open-ended and interesting, encouraging general discussion and imaginative solutions. The presentation of the problem was given early in the course, with an oral description aimed at motivating students to develop ideas for a technical solution.

> **CIS286 Advanced Object-Oriented Techniques**
>
> This course covers application of object-oriented programming to software development which includes the general topics of encapsulation, inheritance, and polymorphism. Topics also include GUI objects, event-driven programming, and exception handling.
>
> **Learning Objectives**:
> 1) Understand the differences between a Java application and a Java applet, and demonstrate the ability to program both.
> 2) Understand and apply OOD principles and OOP concepts to develop Java programs.

FIGURE 1
SYLLABUS SUMMARY FOR CIS286

In our case, we selected a problem faced by many K-8 students dealing with use of flash cards as an instructional tool to support rote-learning and retention. The observation asserted by the instructor, was that grade-school children, particularly those who like computers, find these exercises boring. The example was given about rote-mathematics flashcards (addition, subtraction, multiplication and division facts). Essentially, the motivation for the CIS286 students was to help these grade-school children learn, using a computer to help them memorize basic math facts.

Over the course of several class sessions, the students were quietly placed in the roles of participants, as the instructor facilitated the vision development. Here, all class participants were collectively engaged to create and focus the product vision, with the instructor as a facilitator-participant.

PRODUCT-BASED LEARNING: SELECTING A PRODUCT

Product-based learning requires that the instructor lead in such a way as to allow the design of the product to emerge from the team. This requires facilitation skills in order to develop the details of a product vision shared by the students. Successful management of the product development requires a careful, but imaginative, design process. The essential steps include:

I. Outline the basic product idea.

This is essentially a very informal requirements analysis, where brainstorming is highly encouraged. The goal is typical of brainstorming activities: widen the breadth of scope and information, and then reduce to well-named essential or common elements. For our educational computer-based flashcard product, once the ideas were reduced, the team developed two key observations:

a. Arithmetic flashcards could be more fun if computerized and incorporated into a game.
b. The primary goal was to create an educational tool to promote learning. The product was not to be an educational gimmick that incorporated a few academic elements into unrelated game play.

The instructor then suggested the need for visual performance feedback to support engagement and provide a game atmosphere. From this, the concept of a racecar theme was adopted, and the still-nominal project title of 'Flashcard Racer' emerged.

II. Identify the major issues and questions.

This is an informal requirements analysis step, where the team engages in questioning and role playing to see if the key ideas best meet the product goals. This type of analysis is necessarily specific to the product being developed, as it links issues and goals for further refinement to guide development. Figure 2 presents a summary of the analysis questions linked to product goals for our Flashcard Racer product.

> *Question*: Can we make a flashcard activity fun enough to encourage repetition and practice?
> *Goal*: The game should promote practice and improvement of arithmetic.
>
> *Question*: What motivates students to learn and make progress?
> *Goal*: Students need to be able to see that they are improving.
>
> *Question*: How to promote the learning process and not detract from it?
> *Goal*: The game component should compliment the learning process, not detract from it.

FIGURE 2
CONSTRAINT ANALYSIS: QUESTIONS LINKED TO PRODUCT GOALS

This analysis discussion was a series of questions tied to aspects of the product goal (improve student learning). The end result was that the team came to the following conclusion: *Healthy competition can motivate students to get better at the flashcard activity, thus improving their arithmetic skills.*

At this point in the process, the vision (Flashcard Racer) and, constraints (educational yet motivational through healthy competition among students) were agreed upon by the development team. This product vision and its constraints, including some level of detail, were then available to the team for formulating a conceptual solution.

III. Decide on a conceptual solution

In this phase, the team develops the conceptual design for a product that satisfies the goals and constraints established in the previous step. This includes the development of general use case scenarios that will potentially solve the original problem. The following points summarize key use design components for our specific product as determined through this phase.

a. The product should allow an activity that can accommodate a group of students.
b. The classroom teacher should be able to control and run the activity.
c. Each student should answer flashcards individually.
d. Student flashcards should provide immediate feedback for correct or incorrect responses.
e. A racecar race may be displayed after students are done answering a series of flashcards.
f. Each student has a racecar that moves with a speed determined by their performance on the flashcards.

Figure 3 depicts a concept image showing the collective design decisions made by the development team. This depicts what the students participating in the flashcard exercise/game would experience. Their individual question (configured by the teacher) and answer are located on the left. Their racecar image and random identification number (1) are shown on the upper-left. Ideally, the student would have only one input to type, and their relative 'speed' at completing flashcards would be shown on the speedometer. A design similar to that depicted in Figure 3 was used by the CIS286 students for implementing their flashcard client application in Java.

FIGURE 3
FLASHCARD CONCEPT SCREEN

IV. Develop sufficiently detailed product requirements.

By the end of this phase, the team should be able to lay out a detailed technical product specification that is complete enough to begin development. The primary product requirements for the Flashcard Racer include:

a. Client-server model with the educator as the server and the students as the clients.
b. The teacher starts the activity, sending identical sets of flashcards to all students
c. As students complete their flashcards, their racecar speed is computed based on the number of correct answers and the amount of time taken to complete each flashcard.
d. The client interface will include a speedometer that will display immediate feedback for student responses. Speed will increase for correct answers or decrease for incorrect answers.
e. Results are sent to the server when the student is finished.
f. When all students are finished with their flashcards, the teacher may then start the racecar race if desired.
g. All racecars race across the screen of the educator's computer. Ideally, the visuals could be projected onto a larger screen.

Figure 4 presents the server interface after all student results for a flashcard session have been sent back to the server, and the race has begun. Students are familiar with their personal racecars, but racecars are not identified for other students. This design enables competition while preserving anonymity. Similarly, student data on performance (both speed and accuracy) are saved for later analysis by the teacher.

The resulting design describes a product requiring interworking logical components and in our case, a set of interactive graphical user interfaces. The construction of these components and interfaces became the vehicle for achieving the learning objectives for the CIS286 course (Figure 1). More importantly, this series of product development exercises helped set the tone for an involved learning experience, transitioning the leadership of the product development from teacher-student model to a shared, intentional collaboration, where the product vision is shared by all, and the course instructor is a participant.

FIGURE 4
FLASHCARD RACER COMPETITION SCREEN

PRODUCT-BASED LEARNING: CLASS OPERATION

For efficient class operation, class time was split between traditional lectures and the product specification process during the first several classes. Once the product requirements are in place, the vision motivates the development process. As a part of the team, the instructor is able to offer information that is useful for constructing a sound technical design. The students understand that they must understand both the theory and implementation methods of new material in order to successfully achieve the team vision. Consequently, the students strive to understand the concepts essential to proceed in product development. Though some of this new knowledge is not directly included as learning objects for the course, it is useful new knowledge nevertheless, and it has been attained through the students' personal desires to learn. For example, though not explicit course topics, the Flashcard Racer product required the students to investigate various networking models, object serialization, and educational issues.

Additionally, a thoughtfully selected project problem, along with a thorough design process, will also require sound and complete attainment of the knowledge specified through the course objectives. Our product, for instance, was a networked application involving separate but interacting components (flashcards, question and response data, racecars, racetrack, etc.). A sound understanding of object-oriented design principles helped the students to organize the product components into interrelated classes. Product development not only motivated the students to learn more about object-oriented principles, but also revealed many of the primary advantages to the design techniques through real project experience. For example, students experienced how inheritance and class organization can make code easier to modify and maintain when changes need only be made to specific components of the application.

The organized class design also helped the team delegate responsibilities and efficiently engineer product components. Rather than being told to work on an arbitrary piece of code, students began development by choosing to work on the components in which they were interested. As a result, students took personal responsibility within the product, feeling direct ownership of their components. Working within a close team, these feelings of responsibility can motivate students to succeed as part of a commitment to the group. Furthermore, students are motivated by a responsibility to the end users who hope to benefit from the product.

Though the students originally focused on single product components, they had to collaborate in order to fit the pieces together to create a working system. In addition, later task assignments were determined within the group based on task priority, extent of previous contributions, and knowledge of existing components. This type of development process allowed students to broaden the scope of complimentary topics learned while mandating a mastery of the key principles. This methodology also allowed the instructor to guide development efforts in order to maximize individual learning. Thus, the instructor must be able to monitor individual progress and coordinate component interaction within the team.

To help enforce object-oriented relationships, the BlueJ [9] Java development environment was useful for showing class relationships and dependencies. This visual representation aided the collaborative effort by allowing students to clearly identify interactions between their individual components. Additionally, the unit testing support included in BlueJ was useful for testing the components of the system and otherwise supporting the learning environment for the CIS286 outcomes.

The Java classes (components) were each built and integrated by small teams of students during the course. Students volunteered for various components, and each student or pair of students was graded against the CIS286 course objectives as assignments. In some cases, multiple students or pairs created the same component, and the whole class selected the best design/implementation to integrate into the product. The instructor helped guide the integration of the components, and ensure that the integrity of the product as a whole was maintained.

Because of the open ended nature of the project, the instructor actively works with the students to develop design components that not only satisfy the constraints of the problem, but also allow achievement of the class learning objectives. As unfamiliar topics arise during development, the instructor can explain the material as it relates to the project, evoking examples that are familiar and relevant for the students. These instructional sessions are interjected into the development process as needed. While similar to short lectures, these sessions are closer to discussions. They are engaging because the topic directly relates to current problems that the students must solve in order to succeed in developing their product.

Through leading these sessions, the instructor maintains a degree of control over the presentation of information. The instructor can gauge what concepts are well understood and what areas are less clear. Additionally, these instructional sessions provide common coverage of fundamental materials across the class. This makes it feasible to incorporate traditional tests or quizzes into the course for additional individual evaluations.

PROJECT RESULTS AND DISCUSSION

The project resulted in a fully functioning product that met the design requirements and could be used as a fun learning tool. In addition to checks of individual student contributions, the quality and complexity of the system's design and implementation serves as further evidence for the achievement of the course learning objectives. Student feedback indicated that the project was preferred as a learning experience over traditional, lecture-based methods. Furthermore, observing faculty members and the students noted the high presence of positive attitudes in the class. Based on these results, the project is considered to have been a great success. The product-based method maintained student engagement and encouraged intentional learning tendencies.

An additional benefit was the design and creation of an appealing learning application that saw actual use. CIS286 participants repeatedly commented that the promise of this stakeholder interaction (client testing) and these testing sessions themselves were "rewarding" or "motivating." Informal product testing conducted by Gannon University students and faculty members and their school-aged children made it clear that the Flashcard Racer product can be fun to use, and convert an often boring learning exercise (rote memory) into a motivating experience. Furthermore, even in a limited number of trials, it was noted that the test participants reported that they were sharpening their arithmetic skills. As an informal product test, a group of elementary school students was invited to use the Flashcard Racer application. Our young participants verbally indicated excitement and interest in further use of the software. As a final test of the product design, a group of 15 third grade students were asked to try a cleaned-up version of the software to practice arithmetic. All students reported that they preferred the software to regular paper flashcards, with all but 2 explicitly stating in a questionnaire that it was fun.

The positive application feedback serves as strong evidence that the CIS286 class designed a fun and appealing learning tool, satisfying a major part of the original problem. Though the validity of the claim that the Flashcard Racer is an effective learning tool has yet to be proven, our informal results indicate that the application has enough educational value to satisfy the educational component of the issued problem. For future investigation, it would be interesting to formally study the educational value of the Flashcard Racer design.

CONCLUSION

We have extended the concept of collaborative problem-based learning by focusing on the design and development of a usable product that can be utilized as a real solution. Rather than place the course instructor in a role as an external guide or overseer, our approach is unique in that incorporates the instructor into the development team as an experienced but mutual partner. By giving the team control of the project from the beginning, students take personal ownership of the product, becoming responsible for their own learning. Thus, this approach motivates intentional learning of not only the primary course material, but also supplementary knowledge of topics that help the project achieve success.

The Flashcard Racer project has shown that the product-based approach can successfully motivate students to achieve course learning objectives through a software development project producing a useful product. This method can be applied to other courses related to engineering tools or techniques.

REFERENCES

[1] Barg, M. et al. "Problem-Based Learning for Foundation Computer Science Courses". *Computer Science Education*, 10, 2000, pp. 1-20.

[2] Beck, L. L., Chizhik, A. W., and McElroy, A. C. "Cooperative learning techniques in CS1: design and experimental evaluation". In *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, St. Louis, Missouri, USA: ACM, 2005, pp. 470-474.

[3] Bereiter, C., and Scardamalia, M. "Intentional learning as a goal of instruction". *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, 1989, 361-392.

[4] Boekaerts, M. "Self-regulated learning: where we are today - Theory, research, and practice" , *International Journal of Educational Research 31*, 1999, pp. 445–457.

[5] Daniels, M., Faulkner, X., and Newman, I. "Open ended group projects, motivating students and preparing them for the 'real world'". *In Proceedings of 15th Conference on Software Engineering Education and Training*, 2002. (CSEE&T 2002). 2002, pp. 115-126.

[6] Gokhale, A. A. "Collaborative learning enhances critical thinking." *Journal of Education* 7, no. 1, 1995.

[7] Hiemstra, R. "Self-directed learning". *The International Encyclopedia of Education*. 1994.

[8] Johnson, D. W., Johnson, R. T., and Smith, K. A. "Cooperative Learning Returns to College: What Evidence Is There That It Works?" *Change*, July/August 1998, pp. 27-35.

[9] Kölling, M., Quig, B., Patterson, A., and Rosenberg, J. "The BlueJ system and its pedagogy". *Computer Science Education*, Vol. 13, issue 4, 2005, pp. 249-268.

[10] Martinez, M. "Intentional learning in an intentional world". *ACM Journal of Computer* Documentation (JCD). 24, no. 1, 2000, pp. 3-20.

[11] Rau, W., and Heyl, B. S. "Humanizing the College Classroom: Collaborative Learning and Social Organization among Students". *Teaching Sociology* 18, no. 2, 1990, pp. 141-155.

[12] Sabin, R. E., and Sabin, E. P. "Collaborative learning in an introductory computer science course". *SIGCSE Bull*. 26, no. 1. 1994, pp. 304-308.

[13] Smith, K. A., Sheppard, S. D., Johnson, D. W., and Johnson, R. T. 2005. "Pedagogies of Engagement: Classroom-Based Practices". *Journal of Engineering Education*, Vol. 94, No. 1, 2005.

AUTHOR INFORMATION

Eric Ragan, Center for Human-Computer Interaction and Department of Computer Science, Virginia Tech, ericragan@vt.edu

Stephen Frezza, Computer and Information Science Department, Gannon University, frezza001@gannon.edu

Jeremy Cannell, Computer and Information Science Department, canell001@gannon.edu