Problem-Based Learning and Self-Efficacy: How a Capstone Course Prepares Students for a Profession

□ Joanna C. Dunlap

Problem-based learning (PBL) is apprenticeship for real-life problem solving, helping students acquire the knowledge and skills required in the workplace. Although the acquisition of knowledge and skills makes it possible for performance to occur, without self-efficacy the performance may not even be attempted. I examined how student self-efficacy, as it relates to being software development professionals, changed while involved in a PBL environment. Thirty-one undergraduate university computer science students completed a 16-week capstone course in software engineering during their final semester prior to graduation. Specific instructional strategies used in PBL—namely the use of authentic problems of practice, collaboration, and reflection—are presented as the catalyst for students' improved self-efficacy. Using a self-efficacy scale as pre- and postmeasures, and guided journal entries as process data, students were observed to increase their levels of self-efficacy.

□ Today, many professionals work in a climate of continual change and innovation. To meet this challenge head on and remain competitive in the workplace, professionals need to be content experts, as well as highly skilled problem solvers, team players, and lifelong learners (Dunlap & Grabinger, 2003; Hmelo & Evensen, 2000). This is especially true for people working in fields such as medicine, law, engineering, and software engineering and computer science. In order to prepare people for these fields, educators need to create learning environments that engage students in ways that help them develop content expertise and problem-solving, collaboration, and lifelong learning skills. A problem-centered approach, based on and consistent with the constructivist pedagogical assumption that learning is a product of social and cognitive interactions (Grabinger, Dunlap, & Duffield, 1997; Greeno, Collins, & Resnick, 1996; Spector, 2003), can be used to achieve these development goals.

In problem-centered learning environments, students have opportunities to practice applying their content knowledge and workplace skills while working on authentic, contextualized problems and projects.[1] The terms *problem-centered learning*, *problem-based learning* (PBL), and

1 This type of learning experience or environment is also referred to as student- or learner-centered because (a) it places the learning needs of the students at the center of the instructional design decisions, and (b) the instructional strategies used (e.g., inquiry, collaboration, and reflection) require active participation and drive on the students' part (Grabinger et al., 1997). The problem focuses students' activity, and is used to stimulate student motivation and interest, and encourage them—and, in fact, relies on them—to take more and more responsibility for and ownership of the learning process.

ETR&D, Vol. 53, No. 1, 2005, pp. 65–85 ISSN 1042–1629

problem-centered instruction are often used interchangeably (Spector, 2003) to refer to instructional approaches that use "real world," simulated, contextualized problems of practice to motivate, focus, and initiate content learning and skill development (Boud & Feletti, 1991; Bruer, 1993; Williams, 1993); this "problem first" emphasis is in direct contrast to the more conventional approach of assigning an application problem at the end of a conceptual unit. Problem-centered learning approaches—which include cognitive apprenticeships, case studies, anchored instruction, and intentional learning environments (Dunlap & Grabinger, 2003; Williams, 1993)—tend to share common instructional characteristics beyond the problem-first focus: The approach to learning is context sensitive and situated, and the process students follow replicates the commonly used systemic approach to resolving problems or meeting challenges encountered in the workplace and world at large; the problems that students work on reflect the true nature of the world and are, therefore, complex and ill structured, and without simple, formulaic solutions; students are actively involved in the learning process from problem introduction to solution implementation and process reflection; students set learning goals and create action plans to drive learning activities; students conduct information gathering and research; students reflect on what they have learned and how they have learned; and students work collaboratively with colleagues to pool their knowledge and skills, share the results of their inquiry, engage in peer teaching, and ultimately solve the problem (Barrows & Kelson, 1993; Dunlap & Grabinger, 2003; Grabinger et al., 1997; Koschmann, Kelson, Feltovich, & Barrows, 1996; Spector 2003). Hence, problem-centered, or problem-based, learning environments may help prepare students for their professions because students actually work on problems in ways that require them to develop expert knowledge, problem-solving proficiency, lifelong learning skills, and team participation skills.

One example of a problem-centered learning approach is Howard Barrows's model of PBL—a method refined through use and evaluation at McMaster University Medical School and Southern Illinois University School of Medicine. PBL's learner-centered approach engages students in an iterative, continuous process of building and reshaping understanding as a natural consequence of their experiences and interactions with problems of practice (Barrows, 1984, 1985, 1986, 1992; Barrows & Tamblyn, 1980; Grabinger et al., 1997; Walton & Matthews, 1989). Working in collaboration with peers, students analyze a problem of practice, formulate hypotheses, and identify areas of knowledge gaps to guide research activities (Wilkerson & Gijselaers, 1996). The skills and knowledge acquired through research and study are applied back to the problem; students evaluate the effectiveness of their research activities and the application of their research results to the problem at hand, while integrating their new learning with their existing knowledge. PBL helps students develop competencies that will serve them throughout their professional lives. These lifelong competencies include the ability to (Engel, 1991):

- Adapt to and participate in change.
- Deal with problems and make reasoned decisions in unfamiliar situations.
- Reason critically and creatively.
- Adopt a more universal or holistic approach.
- Practice empathy, and appreciate others' perspectives.
- Collaborate productively in groups or teams.
- Identify personal strengths and weaknesses, and undertake appropriate remediation (self-directed learning and metacognitive skills).

By promoting these competencies, "problem-based learning is apprenticeship for real-life problem solving" (Stepien & Gallagher, 1993, p. 26).

Although PBL helps students acquire the knowledge and skills required of professionals in the workplace, competent task performance also requires self-efficacy (Schunk, 1989b). The acquisition of skills makes it possible for performance to occur, but without self-efficacy the performance may not even be attempted (Mager, 1992).

Self-efficacy is an individual's level of confidence and self-judgment regarding ability to organize and implement actions needed to perform effectively (Schunk, 1989a). Bandura (1977) described the development of a person's self-efficacy perspective as a dynamic process involving self-referent thought, affect, and actions. Research indicates a strong and positive influence of efficacy beliefs on various aspects of student motivation and achievement (e.g., Bandura & Schunk, 1981; Betz & Hackett, 1981; Pajares & Miller, 1994, 1997; Pintrich & De Groot, 1990; Schunk, 1982, 1983, 1984a, 1984b; Zimmerman, Bandura, & Martinez-Pons, 1992; see also Multon, Brown, & Lent, 1991, for a meta-analysis). Schunk's (1982, 1983, 1984a) series of experiments, for example, documented that as students' self-efficacy perceptions strengthened, their performance also noticeably improved.

Instructional strategies, including combining explanations with modeling, giving instruction on specific strategies, setting specific short-term and long-term goals, and giving explicit feedback to individuals about their performance, can serve as sources of efficacy information that enhance self-efficacy development. A number of research reviews (e.g., Albanese & Mitchell, 1993; Berkson, 1993; Colliver, 2000; Davies, 2000; Norman & Schmidt, 1992; Vernon & Blake, 1993) have documented PBL's positive impact on knowledge and skill acquisition and transfer, problem solving, attitudes and opinions about courses and programs, measures of performance, and self-directed learning.[2] There is a lack of research, however, on PBL's impact on student beliefs about their abilities to perform—their self-efficacy. Can PBL—a problem-driven and certainly more student-centered approach that involves different instructional strategies than those listed above—have a positive impact on students' self-efficacy? Because of PBL's use of problems of practice to drive learning, students engaged in a PBL environment should experience performance accomplishments that have an overall impact on their self-efficacy. Using a mixed-methods research approach, I examined how students' self-efficacy perspectives, as they relate to their roles as software development professionals, changed while the students were involved in a PBL environment. The findings from the study were then used to suggest specific instructional strategies that may be incorporated into learning environments to enhance students' self-efficacy.

DESCRIPTION OF STUDY

Participants and Context

Thirty-one students in a 16-week required, undergraduate capstone course in software engineering participated in the study. Using purposive sampling (Patton, 1980) as opposed to an a priori researcher-directed and randomized approach, participants were selected on the basis of their commonness (they were typical cases), and convenience (they were the students registered in the course).

The course instructor and I collaboratively designed the capstone course, selecting PBL as the instructional approach for a number of reasons. We started by conducting a front-end analysis to determine what the course needed to achieve in the program, and what students needed to know and be able to do to meet the expectations of the profession and needs of employers. Then, we considered the types of learning experiences students received leading up to the capstone course and what content and skills students learned during those experiences. This analysis defined a clear gap that the capstone course needed to bridge—primarily, to help students apply what they had learned in their more didactically oriented courses to professional problems of practice. To achieve this, the course instructor and I investigated the problem-centered approaches listed previously, including PBL. We determined that the PBL approach was appropriate for use with com-

2 The positive results reported by many researchers are not universally accepted. Some researchers have raised concerns that comparative research on the effects of PBL has not presented conclusive results on gains in expertise as compared to more conventional approaches (e.g., Hmelo, Gotterer, & Bransford, 1997; Norman & Schmidt, 2000). Others have questioned the positive results because many PBL studies lack clarity about given approaches to PBL and the way researchers reviewed the learning outcomes (e.g., Kelson & Distlehorst, 2000). In addition, there are also questions about its lack of clearly detailed theoretical foundation (Spector, 2003), and standard for guiding PBL practice and empirical research (Koschmann, 2001).

puter science students because it closely aligned with the software development life cycle (SDLC) approach to software engineering problems; we believed that this close alignment would help reinforce student use of the SDLC approach to problem solving—an approach followed in one form or another by most software engineering professionals—instead of introducing a new approach.

The capstone course, which occurs in the last semester of the computer science degree program, was the students' first exposure to PBL. This course provided an opportunity to use PBL at a critical time in their program—a time when they could apply their experience and knowledge from all previous courses to practice their careers. The PBL project designed for this course involved students in a significant systems design project with an actual client. Reflecting a typical software development team structure, students worked in groups of three or four to define the client's problem and associated needs, write proposals, conduct analyses, design solutions, and implement and test those solutions. To launch the PBL experience, students received a request for proposal (RFP) from the client, for which they submitted a proposal. They then actually conducted a detailed software analysis, developed a software design solution, and implemented and tested their software solution. Courses leading up to the capstone course did not employ PBL techniques; hence, student growth in self-efficacy as a result of participating in PBL could be observed directly.

For this semester-long project, the problem students had to address involved a request from a construction company to develop a Web-based project-planning application. To work on this problem, students had access not only to the RFP but also to materials typically available at an actual information technology (IT) development firm, such as the firm's software development methodology and philosophy, coding standards, and testing standards. As the project progressed and students interacted with the client to draw out more information, they were furnished with additional materials, such as samples of existing client reports that needed to be automated, and descriptions of existing workflows within the construction company. In order to address the construction company's problem, the students had to complete four tasks during the semester: (a) respond to the RFP, (b) conduct a detailed software analysis, (c) develop a software design solution, and (d) implement and test their software solution. Each of these tasks had specific deliverable products that students had to produce and submit for client review. For each of these tasks—each representing a problem that students had to solve—students went through the four phases Barrows (1985) prescribed for a PBL activity:

- *Phase 1:* Reasoning through the problem and identifying educational needs in counterpoint.
- *Phase 2:* Engaging in self-directed study.
- *Phase 3:* Applying new knowledge to the problem and critiquing prior problem work in counterpoint.
- *Phase 4:* Summarizing and integrating learning.

In order to better reflect the SDLC, Barrows's four phases were renamed as follows: (a) problem analysis (Phase 1), (b) solution design (Phase 2), (c) solution development (Phase 3), and (d) postdevelopment review (Phase 4)[3].

Thus the first iteration involved students going through these four phases in order to respond to the RFP and develop a proposal. The second iteration involved the same four phases, but to conduct a detailed software analysis. The third iteration was for developing a software design solution, and the fourth iteration was for implementing and testing their solution (see Figure 1). In fact, as described earlier, one of the reasons PBL was originally selected as the instructional approach for this course was because Barrows's phases closely reflect the SDLC and, therefore, the types of activities software engineers complete during a development project. Because this project had four specific tasks to complete—or problems to solve—students had four opportunities to follow the PBL

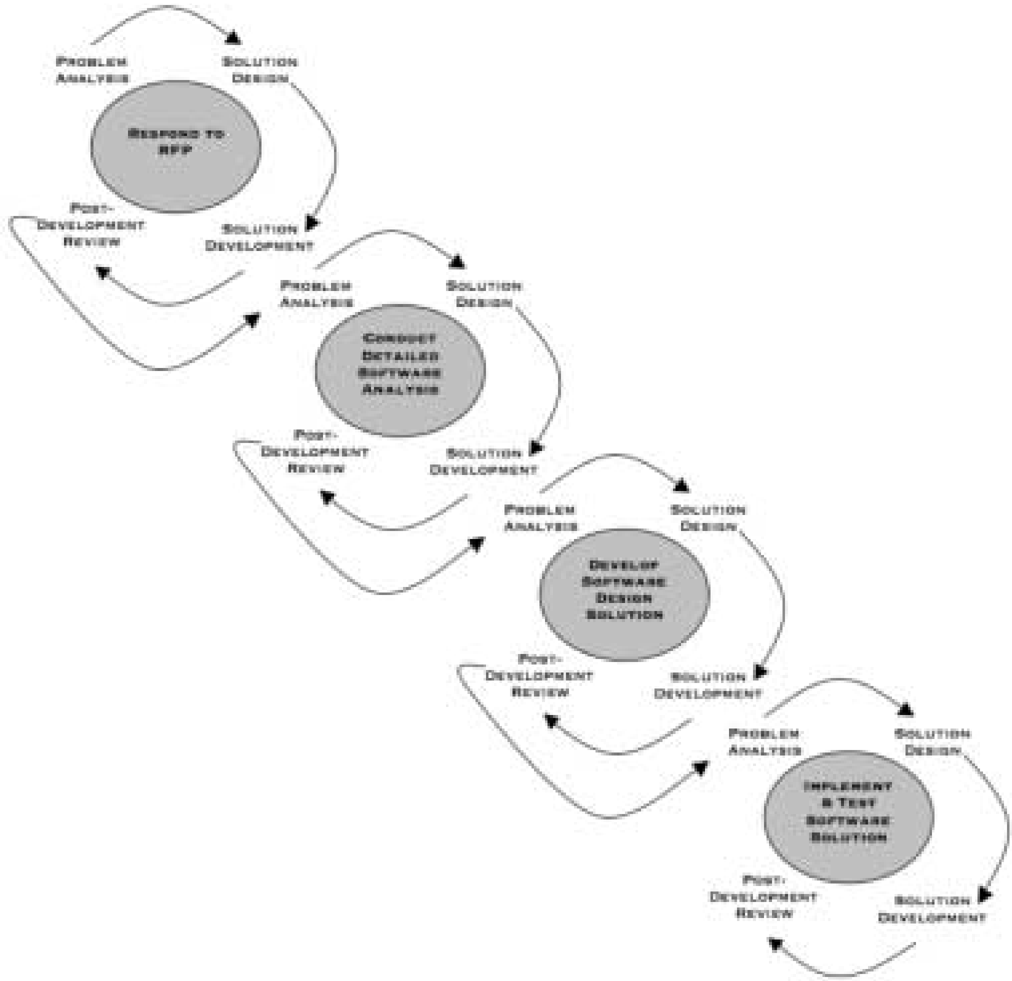
3 See Table 1 for an illustration, using the specific activities of the respond-to-RFP task, of the connection between Barrows's PBL phases, and the activities students did during each capstone project task.

Table 1 ☐ Connection between Barrows's problem-based learning (PBL) phases and student activities during the respond-to-RFP task.

Barrows's PBL Phase	*Renamed Phase to Align with SDLC**	*Activities According to the Literature*	*Activities During the Respond-to-RFP Task*
1: Reasoning through the problem and identifying educational needs in counterpoint	1: Problem analysis	Students are formed into groups (or one large group). Problem is presented. Students evaluate problem: Students offer ideas and hypotheses, significant facts learned, and information that is still needed. Students determine what they need to do next to work on the problem and create action plans to complete the inquiry.	Students are formed into teams. Problem is presented via RFP from client. Students evaluate the RFP: Students offer ideas for addressing the client's needs, significant facts that they know based on the RFP and on what they have learned about similar problems in other courses, and information that is still needed to be able to address the client's requirements as presented in the RFP. Students determine what they need to do next to work on the RFP and create action plans to complete the inquiry.
2: Engaging in Self-directed study	2: Solution design	Based on action plans, students engage in self-directed study. Students decide on the best learning resources for this inquiry. The problem solving of Phase 1 facilitates student processing of information they obtain during inquiry.	Based on action plans, students engage in self-directed study to gather information that will help them respond to the requirements in the RFP. Students decide on the best learning resources for this inquiry. The problem analysis of Phase 1 facilitates student processing of information they obtain during inquiry.
3: Applying new knowledge to the problem and critiquing prior problem work in counterpoint	3: Solution development	Students return from the self-study period as experts with the information needed to solve the problem. Students critique learning resources used during self-study. Students apply their newly acquired knowledge to the problem. The group decides what ideas should be considered, and what next steps should be completed. When the group believes it has enough knowledge to solve the problem, it implements a solution.	Students return from the self-study period as experts with the information needed to respond to the client's RFP. Students critique learning resources used during self-study. Students apply their newly acquired knowledge to their RFP response. The group decides what ideas should be considered for the RFP response, and what next steps should be completed to gather more information to finish the RFP response. When the group believes it has enough knowledge to write a response to the client's RFP, it finalizes the RFP and submits it to the client.
4: Summarizing and integrating learning	4: Post-development review	Students reflect on and summarize what they have learned.	Students reflect on and summarize what they have learned. This is done during a structured walkthrough of the RFP response.

* The phases were renamed for the students' benefit to reinforce PBL's alignment with the SDLC problem-solving approach.
Note: Phases 1–3 are iterative until the group believes it has the knowledge it needs to proceed.

Figure 1 □ Relationship between the four phases of problem-based learning (PBL) and the project tasks. Students complete the four phases for each project task. The results of each four-phase iteration inform the next project task.



approach, practicing problem analysis, collaboration and teamwork, inquiry, and reflection skills.

For example, after the RFP–proposal process had concluded with the postdevelopment review (Barrows's Phase 4), problem analysis (Barrows's Phase 1) began for conducting a detailed software analysis. The instructor—in his role as tutor—created four columns on the board with the following headings: (a) Ideas and Hypotheses, (b) Facts and Problem Information, (c) Learning Issues, and (d) Action Plan (again, this same activity had occurred earlier when students were asked to respond to the RFP). Although selected in advance, the teams worked in unison to conduct the problem analysis to ensure that all teams recognized the same client problem and learning issues. Using the columns on the board as a guide, the instructor guided examination of the student proposals along with the IT firm's standards in such a way as to begin drawing out a detailed software analysis of client requirements. Students quickly generated ideas on what a completed detailed software analysis might look like ("We will have to generate use cases and a prototype") and factual information from the earlier proposal process ("The application will need two levels of security—one for normal users and one for administrative users"). As the session progressed, students identified learning issues that would require additional study ("What is the difference between a storyboard and a prototype?"). Finally, the students developed an action plan to begin their detailed software analysis. Functioning as a PBL tutor, the instructor did not provide direct instruction about the project or how the students should proceed, but subtly guided students into appropriate discussions by asking metacognitive-level questions such as:

What do you notice about the client's request for that functionality?

What information is important in the documentation provided by the client?

Do you have any ideas about dealing with the two levels of security?

Why don't you summarize the important facts we've learned so far about the client's requirements?

The instructor also helped students move through the process of generating and prioritizing ideas and making decisions about next steps, and checked for consensus whenever the group made a decision.

During the second phase of the conducting-a-detailed-software-analysis task—solution design (Barrows's Phase 2)—students collaborated in teams of three or four to outline the content of their detailed software analysis while adhering to the previously generated proposal, and rules and constraints identified in the IT firm's standards. To do this, they first carried out the action plan developed during the problem analysis phase by engaging in self-directed study, using a variety of resources including books, online help, experts, and Internet resources. During this phase, the instructor encouraged students to expand their resources beyond using their textbooks, and pushed students to study topics of which they were less knowledgeable. After conducting their research and synthesizing their findings, students regrouped in their project teams and, acting as experts on their researched topics, applied their newly acquired knowledge to the problem.[4] The instructor monitored students to make sure they were finding and applying research that helped the group solve the client's problem, employing the same questioning techniques used during the first phase.

During the solution development phase (Barrows's Phase 3) of the conducting-a-detailed-software-analysis task, students worked together to apply the results of their research to developing an actual detailed software analysis for the client's problem. Throughout this process of applying their research to the problem, students revisited the four columns of information to (a) test their ideas and revise their hypotheses, (b) generate additional ideas and hypotheses, (c) verify or modify their understanding of the problem, (d) identify new learning issues, and (e) adjust their action plan to complete their work. They added these modifications to the original four columns. During this time the instructor went around the room offering support and making sure that students stayed on task. Class meetings were also used as opportunities for students to discuss problems they encountered and to share the solutions they had developed to solve those problems.

The final phase—postdevelopment review (Barrows's Phase 4)—happened once students finished conducting their detailed software analysis. During this review, students discussed the analysis techniques they had used, shared solutions and discussed possible alternatives, and addressed any lingering questions. Students also used this time to reflect on the process itself, discussing what learning strategies they had employed, what strategies and resources worked and did not work, what they would do differently in the future when analyzing a software project, and how they would improve their team process. As each student shared a self-evaluation, the instructor encouraged other students to make additional comments about each student's performance, and shared his own thoughts regarding what was good and what

4 This itself often takes several iterations of considering the action plan, conducting research, sharing research findings, and conducting more research based on new learning issues that come up.

could be improved for the future.

The project progressed through the other project tasks, following Barrows's model faithfully. By following Barrows's PBL structure, the students participated in a project that incorporated the essential characteristics of PBL (Kelson & Distlehorst, 2000):

- Students were involved in small group collaborative learning.
- Students were aided by a tutor who probed them with questions that lead them toward conceptualizing and solving the problem, by coaching the teamwork process, and by facilitating the learning process without providing information or answering questions about the problem or content.
- Learning was stimulated through use of an actual workplace problem.
- Students were required to inquire for further information, to engage in self-directed learning, to employ hypothetico-deductive reasoning, and to interact with group members to benefit from their perspectives in constructing an understanding of the problem and its solutions.

Data Collection and Analysis

This study used a nonexperimental, single-group research design to describe student experiences. To examine the process of change by comparing the group's performance or level of change to a baseline level, students completed a guided journal submission and the General Perceived Self-Efficacy Scale before the PBL experience commenced. The results from these pre-PBL data collection activities provided baseline measures of student self-efficacy. A time series approach to collecting students' guided journal responses mitigated the impact of other factors on the study's findings, allowing for comparisons in student self- efficacy at different points during the PBL experience. A pretest-posttest approach was followed for the General Perceived Self-Efficacy Scale.

Guided journals. Journaling is "a method of promoting exploration and facilitating reflection on learning and new experiences within the context in which the learning unfolds" (Gillis, 2001, p. 49). Journal writing fosters understanding and the application of concepts (see Connor-Greene, 2000), enhances critical thinking (see Hettich, 1990; Hodges, 1996), and improves achievement and attitude (see Borasi & Rose, 1989; Jurdak & Zein, 1998). It is also a powerful research tool for capturing students' reflective practice, conceptual change, thinking, and learning. Moreover, because journal writing serves valid instructional and research purposes, the data collection may intrude less into students' authentic learning experiences (although knowing that they are participating in a research study may lead to a Hawthorne effect).

For this study, students were required to complete a reflective journal at specific points throughout the duration of the course. This journal was due once every three weeks, leading to a total of five completed journals for the semester. Students received a set of questions each week to guide their journal responses. They sent their journal responses to me (not the instructor) via e-mail. The journal responses were not graded by the instructor, although the students were required to submit their responses as part of their class participation requirement; this requirement fit well with the PBL experience because PBL emphasizes the importance of reflection as part of its structure and process.

A caveat to utilizing journals as data collection instruments is the possibility that some of the changes in the quality of students' journal writing over the course of the project are caused by acquiring journaling experience via practice, and gaining a better understanding of expectations. However, before data collection commenced, students submitted a practice journal after the first week of the semester (the first week was focused on review activities, with the PBL project starting in the second week). This activity addressed the same questions used for data collection. This practice activity minimized the impact of practice on the quality and detail of journal responses. Throughout the actual data collection, students received no clarification or feedback regarding the quality or quantity of their journal responses.

The questions used to guide student journaling were designed to capture self-efficacy perceptions by asking students to reflect on their ability to organize and implement the actions needed to perform effectively in the software development profession (Schunk, 1989a) at different points during the PBL experience. Specifically, the questions encouraged students to comment on their confidence with regard to their software development knowledge, skills, and dispositions; preparedness to work on software development projects; readiness to meet the real needs and demands of the profession; and ability to function as software engineers. For example (see Appendix A for the full set of questions):

- What did you learn about your ability to work as a software development professional (analyst, designer, programmer, project manager) over the last three weeks?
- How well do you think you would do if a software development firm suddenly hired you? What skills and knowledge do you have to contribute? What would you still need to work on?
- Update your resume. What did you learn about your abilities to work as a software development professional (analyst, designer, programmer, project manager) over the last three weeks that should be added to your resume?
- Are you confident that you can deal with the demands of real software projects? Why or why not?
- Compared to three weeks ago, how confident are you that you can deal with the demands of real software projects?

Journal responses were examined to determine student levels of self-efficacy prior to PBL participation, their levels of self-efficacy while engaging in the PBL activity, and those aspects of self-efficacy that were improved or enhanced throughout the PBL project. To do this, the first journal assessed students' pre-PBL perceptions (baseline of student self-efficacy before participating in a PBL environment). The second through fourth journals determined students' perceptual changes during their PBL experience, and the last journal determined students' perceptions after participating in a PBL experience.

I established my coding scheme by focusing on general coding categories pulled from the literature: changes in confidence regarding software development abilities and changes in professional identity. Using a coding table of categories with examples, two coders (a colleague and I) first reached consensus on a subset of student journals, and then coded the journals independently to estimate interrater reliability using Cohen's kappa, a measure of percentage agreement corrected for chance agreement. We achieved a Cohen's kappa of 86% agreement.

General Perceived Self-Efficacy Scale. To triangulate the results of the guided journals, students also completed the General Perceived Self-Efficacy Scale (Jerusalem & Schwarzer, 1992; Schwarzer, 1993; Schwarzer & Jerusalem, 1995; Schwarzer, Mueller, & Greenglass, 1999) at the beginning and end of the PBL experience. The General Perceived Self-Efficacy Scale (also known as the Generalized Self-Efficacy Scale) is a 10-item psychometric scale that assesses optimistic self-beliefs to cope with a variety of difficult demands in life. The scale typically yields internal consistencies between Cronbach's alpha = .75 and .90 (Schwarzer, 1997). "The scale is not only parsimonious and reliable, it has also proven valid in terms of convergent and discriminant validity. For example, it correlates positively with self-esteem and optimism, and negatively with anxiety, depression and physical symptoms" (Schwarzer, 1997, Method section, ¶1). In contrast with other scales designed to assess optimism, this scale explicitly refers to personal agency, that is, the belief that one's actions are responsible for successful outcomes. Using a four-point scale—ranging from 1 = *not at all true* to 4 = *exactly true*—the scale aims at a broad and stable sense of personal competence to deal effectively with a variety of problem-solving situations (see Appendix B), with questions such as:

- I can always manage to solve difficult problems if I try hard enough.
- I am confident that I could deal efficiently with unexpected events.
- I can usually handle whatever comes my way.

To reduce contextual ambiguity and assist students in completing the scale, students considered their responses to the scale from the perspective of solving software development problems.

FINDINGS FROM THE GUIDED JOURNAL AND GENERAL PERCEIVED SELF-EFFICACY SCALE

I present the guided journal findings by coding category: changes in confidence regarding software development abilities, and changes in professional identity. Within each category, I describe student perceptions at three different time intervals: pre-PBL perceptions (at the beginning of the course), during PBL perceptions (during the course), and post-PBL perceptions (at the end of the course). The journal excerpts presented are samples of student journal responses (note that no grammatical corrections were made to the responses). The excerpts are representative of the perceptions shared by all of the students in the course; all students expressed concerns about their software development abilities and role in the profession at the beginning of the course, and described improved confidence about these things at the end of the course.

Guided Journal Findings

Changes in confidence regarding software development abilities. The guided journal responses reflected dramatic changes in student confidence regarding software development knowledge, skills, and dispositions, and ability to contribute to software development projects, from the start of the course to the end of the course. At the beginning of the course, 29 students expressed concerns about whether they were prepared to be software developers in the real world. For example, 20 students reported their need to learn more before they could work in the real world:

> My ability to work as a software development professional is very weak. I think I need to learn more and more for each step such as consulting with the client, taking down the requirements, making analysis, etc.

> I've discovered that I really know nothing. I mean in the previous course we didn't have to do anything similar to what we have to do in this course [work on a real project].

> I am not confident at the current time, because I don't think that I know enough . . . I am not sure if I can deal with the real demands of a real project, and that is because I still need to learn more and I have just started learning about real projects.

The other nine students focused their journal responses on their lack of experience in software development, including using various programming languages and working on a project team:

> At this point . . . I would not feel good about being hired as a software developer. I haven't had experience with "real" software projects, so there's quite a bit of uncertainty as far as what kinds of problems can arise.

> I am very inexperienced—plain and simple. I don't have a broad range of programming language knowledge, and I haven't legitimately dealt enough with the intricacies of working on a software project.

> If I got suddenly hired, there are few things that I would be very uncomfortable with and these things are: lack of self-confidence, not enough knowledge about programming, I don't know how to work on a project or on a team.

Seven students even recognized the importance that confidence plays in being an effective software engineer:

> The thing that would make me most uncomfortable if I was hired now is that I am afraid that I could not do my job or have confidence that I know what is expected to complete any job presented to me.

> Whether or not I will be able to deal with the demands of real software projects, perhaps, depends on difficulty of a project and how comfortable I feel in the field it relates to (web design, databases, writing a software, etc.).

> Basically, I'd feel inadequate and uncomfortable . . . . I just don't have the confidence needed to be successful I guess.

The PBL environment immersed students in a real world software development project throughout the semester. During this time frame, students completed three reflective journals. While working on the construction company project (which involved responding to an RFP, conducting a detailed software analysis, designing a software design solution, and implementing and testing the software solution), 30 students began expressing changes in their self-confidence with regard to software development:

I am not 100% confident about dealing with the demands of real software projects yet, but I think I am getting more experience with real software projects to have more confidence because of this class.

The difficult work that we encountered and were able to get a solution for over the last few weeks give me more confidence that I alone or with group of people will be able to tackle a real software projects.

I feel more confident because I learn more and more as we move on to different stages of the project. But I also feel scared because the more I learn the more I feel how little my knowledge was. There are a lot of things I need to learn and practice more.

Twenty-seven students, in fact, reflected on how they were gaining experience during the course that would help them on the job:

I believe that after I finish the analysis and design phase I will learn/gain some real experience to help me in the real world.

I've learned a lot over the last three weeks. At first I thought this class is just similar to the Software Engineering class where we only needed to build a software project we liked. This class is more like in a real world.

I've learned some skills to work in a real world over the last three weeks. However, I still do not feel very confident if suddenly hired.

As I learn more and more, I feel I can deal with harder and more difficult demands of a real software project.

Although more confident, a couple of students still expressed concerns about being effective in an actual job:

50% better in terms of confidence, but I believe I have a lot to learn still before working in the real world.

Compared to 3 weeks ago, I think now I am much more confident because of my accomplishments with the class project. But . . . I still don't know if I would be great on the job.

By the end of the semester, students expressed very different views of their abilities to contribute to software development projects. Students' participation in a PBL environment seemed to have an impact on their personal appraisals of capability, specifically in regards to their abilities to be software development professionals. The students recorded these positive views in their journals after they had submitted their final projects but before they received project feedback or grades from the instructor. (In this way, students' responses were not biased by the grade and feedback received on their final projects.) Twenty-seven students indicated that because of the course they were ready to deal with the demands of actual software development projects, even though some seemed surprised about their newfound confidence:

After the completion of the class, and finishing our team project I feel I can handle any software project!

I'm sure I can tackle ANY [software development] problem. A resounding YES [I am confident I can deal with the demands of real software projects] is the only response.

I never thought I would be but, yes, I am confident that I can adapt to the demands of real software projects.

A few of these students were very specific about what they were now confident about:

The further knowledge that I have gained in the analysis process gives me confidence that if I was suddenly hired, I can analyze a client and define his or her problem with ease.

In addition, 24 students expressed an enthusiasm about their abilities to work in software development, and their desire to get jobs that allow them to do that work.

I feel that if I was hired right now by a firm that I'd be able to go in there and get right to work. I'm definitely ready.

Finally! I now feel really comfortable in my ability to work in software development. I really think I have marketable skills now. My resume looks great!

Yes, I am confident that I could deal with the demands of real software projects. I've been getting more experience with this class for one, and now I'm ready to get out there and find a great job.

In fact, 12 students shared that they did not realize how enjoyable software development was until this course:

Yes, I am confident. I understand the processes and the related difficulties. I feel like I can break the problems down and deal with them. Also I really enjoy working with these software projects, and I didn't really know that before this class.

Changes in professional identity. The journal responses collected throughout the course revealed changes in the ways students described their professional identity and their roles as software engineers. In fact, after the PBL experience, students started referring to themselves as software developers rather than students, suggesting they were confident about their abilities to perform the workplace tasks required of software development professionals.

At the beginning of the semester, 25 students indicated that they did not identify themselves as software engineers, offering comments such as, "I am just a student and still don't see how I am ever going to be ready to be a software engineer" and "Right now I would be in trouble as a programmer. . . . I'm far from being any good at this." However, these perceptions shifted while working on their projects during the course. Students started to refer to themselves by professional titles, although still expressing concerns about their ability to contribute professionally. For example, three students shared:

I feel worse about my programming abilities than I have ever felt before. I would never make it as a programmer at this point!

I am a terrible programmer. If I was hired by a development firm, I would be doing a terrible job.

My abilities as a software developer is improving but I wouldn't be completely confident if suddenly hired. . . . I don't think I know much in depth. I'm still in the process of learning and gaining experience.

By the end of the semester, 28 students were describing their readiness to work as software engineers:

I would be very comfortable if hired as a software developer.

I have found that over the past few weeks my confidence as a software developer has improved. I feel more confident that I know what I am doing and that I am able to perform what is required for this project and that my project will turn out well.

I feel a lot more confident in my development skills. If I was suddenly hired I would feel confident about starting as a software developer.

Even more positively, 16 students referred to themselves as software engineers, as opposed to students without the knowledge and skills needed to contribute to real world projects:

I have gained a lot of confidence as a software engineer.

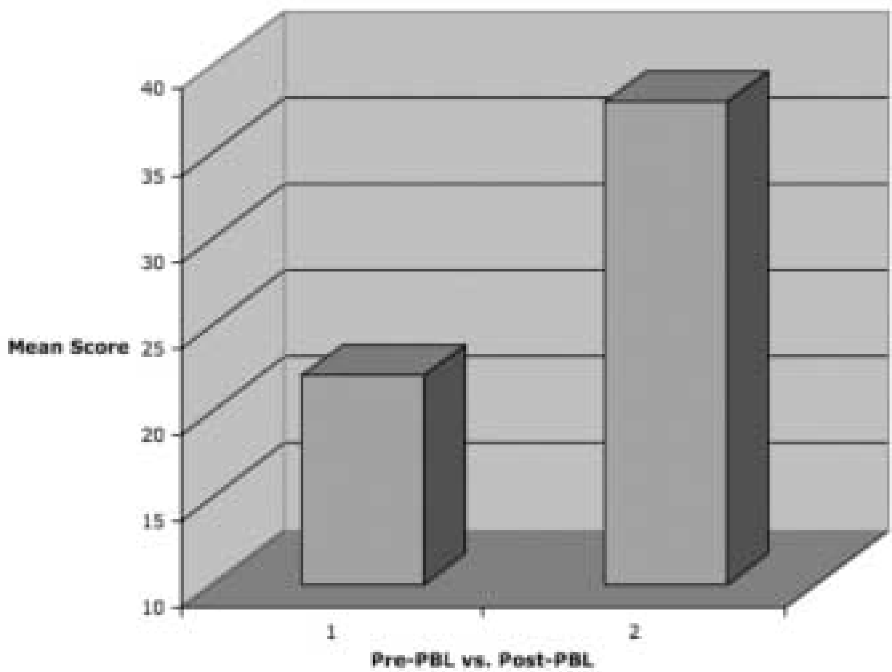
I am very confident that I can now deal with the demands of real software projects because for me, I now know that I know how to be a software developer. I am a software developer.

General Perceived Self-Efficacy Scale Findings

Students completed the General Perceived Self-Efficacy Scale (see Appendix B) at the beginning of the semester (pre-PBL) and at the end of the semester (post-PBL). The 10-item sum score had a theoretical range from 10 to 40, due to the 1-to-4 response format. The pre-PBL mean was 22.07 (SD = 4.553, N = 31). The post-PBL mean was 37.90 (SD = 2.574, N = 31). A two-tailed, paired, dependent t test determined pretest-posttest differences in the group's General Perceived Self-Efficacy scores. The mean self-efficacy scores increased significantly from the pretest to the posttest [$t(30\ \text{df}) = -27.878$; $p < .0001$].

The results, consistent with the journal responses, indicate a significant positive change

Figure 2 ☐ Mean score gains on General Perceived Self-Efficacy scale from pretest to posttest. (PBL = problem-based learning.)



in student perceptions of personal ability and preparedness for the software development profession.

DISCUSSION

To prepare for the workplace, students must have confidence in their ability to perform, and believe they can be successful; students are in a better position to put what they have learned into practice when they have self-belief (Manz & Manz, 1991). The guided journals show a change in almost all (29)[5] of the students' perceptions about their abilities to be software development professionals—from a lack of self-efficacy before their semester-long PBL experience to confidence at the end of the semester. The dramatic increase in the group's scores on the General Perceived Self-Efficacy Scale supports the journal findings. These data were collected after students had submitted their final projects to the client, but before they had received final course grades. To determine final grades, the client worked collaboratively with the course instructor to complete evaluation forms rating student projects in terms of thoroughness, sensitivity to client needs and requirements, professionalism in communicating with the client and presentating products, sophistication and creativity of approaches and solutions, and appropriateness (in terms of addressing the client's requirements within the scope, timeline, and resource availability) of products. The client involved in this study remarked on how impressed he was with the students' work. All students in this course received a grade of *B* or higher, based on client evaluations, so students' perceived achievement was in line with their actual performance. Through their participation in the PBL environment, students ultimately saw themselves as

5 The journal responses from 2 of the 31 students were not detailed or specific enough to determine whether the changes in their perceptions were significant.

software development professionals who were ready to perform effectively in the workplace. Improved self-efficacy can come from four sources: (a) vicarious experiences, (b) persuasive statements, (c) physiological states–arousal, and (d) performance accomplishments, with performance accomplishments, such as successful task completion, having the highest impact on an individual's self-appraisal of capability (Bandura, 1977, 1986). The PBL environment—with the use of authentic problems of practice, collaboration, and reflection—provides students with vicarious experiences and performance accomplishments.

Vicarious Experiences and PBL

This study's PBL environment encouraged collaboration to provide students with opportunities to see and hear how other students approach and solve problems. Because students are working collaboratively during problem solving, their thinking processes are observable and therefore open for personal and peer assessment and refinement. Students test ideas, identify misconceptions, and challenge each other's thinking (Johnson & Johnson, 1979; Lowry & Johnson, 1981). Group participation means that the members must understand many different roles and viewpoints to gain additional insights into the problem (Grabinger et al., 1997). This sharing engages students in vicarious learning, allowing them to note the consequences of peers' activities and learn from their experiences. In addition, PBL's collaborative process provides explicit feedback to students about their performance, serving as a source of efficacy information that enhances self-efficacy development.

Performance Accomplishments and PBL

By working on an authentic software engineering project that reflected real problems of practice in the profession, students had the opportunity to complete relevant, meaningful tasks (Boud, 1985; Boud & Feletti, 1991), which led to performance accomplishments. In a PBL environment, students often take on roles that require authentic decision making, such as engineers, policy makers, school board members, or executive boards. These roles help lend authenticity to the problem, for learners must not only solve the problem, but also act consistently within the role that they play. Through authentic activities, students have an opportunity to practice applying knowledge and skills to new and novel problems, and participate in the community of practice (Albanese & Mitchell, 1993), improving their ability to transfer their knowledge and skills to future challenges (Grabinger et al., 1997). Twenty-two students reported that they found the instructor's use of authentic problems of practice very effective, for example:

> I know I can deal with demands of a real software project. I think it's a matter of having enough motivation, knowledge and experience—which is exactly what I got in this class.

> Yes, I am confident because of technique of the class, that we worked on a real project.

> Most of my strengths in software design come to me because of this class. I feel like I got some true work experience because of the way you set up the class.

Additionally, through the PBL process, students set goals based on learning issues, created action plans to achieve those goals, and identified and evaluated appropriate learning strategies and resources (Barrows, 1985). These activities helped students structure and complete tasks—making it more likely that they would effectively accomplish the goals. Because of how well PBL phases map to the stages of the SDLC, students were better prepared for the types of cognitive activities that occur at each step:

> I now have a much better idea of how to approach the problem and what to start from.

> If suddenly hired, I feel that the last three weeks helped me kind of build the structure of how a project should be done—first spend some time analyzing what it is we have and what needs to be done, without yet worrying too much of how it will be done and languages that I'll use. Then, the step of developing, when I'm deciding how and what I'll be doing, what tools I'll

be using, etc. The final step is implementing/coding itself.

I think after the last few weeks, I am much more confident on problem analysis/break down as well as design. I now understand how this process works, and what I am supposed to be doing at each point in the project.

Reflection—a critical activity in PBL—reinforces students' sense of accomplishment. During the summary-and-integration-of-learning phase of PBL, students reflect on and summarize what they have learned, and discuss its use with future problems. They work out contingencies, explore alternatives to their solutions, and determine ways to improve both their solution and the process they used to arrive at that solution. Students consciously recall and reflect on the learning that occurred while they were solving the problem, elaborate on that learning, and integrate it into their existing knowledge structures (Barrows, 1985). Because PBL focuses student attention on their learning processes and what they have achieved, this activity reinforces their performance accomplishments.

Through authentic activities, collaboration, and reflection, the PBL experience gave the computer science students in this study the opportunity to engage in vicarious learning and increase their performance accomplishments by successfully working through a complex software development project. From a design perspective, the use of these instructional strategies in a learning environment may contribute to enhanced self-efficacy by improving student perceptions of their ability to perform effectively and their professional identity.

CONSIDERATIONS FOR FUTURE RESEARCH

At a macrolevel, some directions for future research include further examination of the relationship between perceived self-efficacy and actual achievement, and the accuracy of student self-reporting as a predictor of performance; the development of a clearer, *practice-based* description of PBL—or what people really *do* when they are *doing* PBL (Koschmann, 2001)—and elaborated theoretical foundations describing the relationship between learning and instruction in PBL environments that can be used as a guiding framework for PBL practice (Hak & Maguire, 2000; Koschmann, 2001); and conducting empirical research that describes the effect of PBL on higher order learning outcomes (Spector, 2003).

At a more microlevel, based on the results of this study, the two most significant areas of focus for future research are (a) the impact of journal writing on students' self-efficacy, and (b) the impact of PBL on students' enculturation into the software engineering community of practice.

Impact of Journal Writing on Self-Efficacy

The journals helped students focus on their perceptions of personal preparedness for the demands of the software engineering profession and workplace. Even though employed as a research method, the guided journal writing engaged students in the type of reflective activity that puts them in a better position to translate theory into practice (Argyris & Schön, 1987). In this study, journal writing may have functioned as an instructional strategy that gave students an opportunity to reflect on and articulate their performance achievements, possibly precipitating change because "simply to record our behaviour is to interfere with it" (Simons, 1978, p. 18). So, the possible connection between journal writing and the changes in student self-efficacy requires investigation in order to determine to what extent—if any—journaling had on the results described in this study.

Impact of PBL on Enculturation into the Community of Practice

Enculturation is the process by which individuals learn about and identify with their domain culture. According to Brown, Collins, and Duguid (1989, p. 34), "the activities of a domain are framed by its culture." However, conventional schooling tends to expose students only to the culture of the classroom, not to the culture in which the content and skills they are learning are naturally applied. Although students are exposed to the tools of a domain's culture during their academic careers, that exposure can be somewhat antithetical to the real requirements of successful professional activity within that domain. Therefore, students may develop knowledge, but because it is inert knowledge (Whitehead, 1929), it may be difficult to transfer it to professional activity.

Studies have shown that becoming a professional occurs on two levels (Hall, 1987; Kerr, VonGlinow, & Schriesheim, 1977). First, it takes place on a structural level, such as formal educational and entrance requirements for entry into the profession. Secondly, it occurs on an attitudinal level, such as the individual's sense of a "calling" to the field. Stated another way, people entering a profession experience change externally, which is in the requirements of the specific career role, and internally, which is in the subjective self-conceptualization associated with the role (McGowen & Hart, 1990).

Preparing individuals to apply theory to professional practice, professional enculturation is a social learning process that includes the acquisition of specific knowledge and skills that are required in a professional role, and the development of new values, attitudes, and self-identity components (Hall, 1987; McGowen & Hart, 1990; Watts, 1987). Through their participation in a PBL environment that required them to take on the role of software developers and practice solving problems that professionals face in the real world, students learned to use the knowledge, skills, and tools of the domain as professionals use them. Acting as practitioners and using the culture's knowledge, skills, and tools to address authentic problems exposes students to the culture of expert practice (Brown et al., 1989). In addition, these activities may also affect students on an attitudinal level, changing the way they identify with the profession. Examining the journal responses over the duration of the PBL experience, students progressed from not identifying themselves with a professional community of software developers to referring to themselves as software developers. Therefore, PBL may influence students' professional enculturation because "drawing students into a culture of expert practice in cognitive domains involves teaching them to think like experts" (Collins, Brown, & Newman, 1989, p. 488). Beyond this study's design and conceptual framework, the influence of PBL on students' professional enculturation would be an interesting area to further investigate. In any case, more attention should be given to instructional methods that support this important process of professional enculturation.

RECOMMENDATIONS FOR DESIGN

Incorporating a PBL approach, whether Barrows's model or another problem-centered model, is an important design consideration for anyone trying to achieve improved student self-efficacy with regard to domain-specific abilities, performance, and professional identity. The PBL environment described in this study gave students the opportunity to increase their performance accomplishments by actually working through a complex software development project. Including similar opportunities for students in professional academic programs is critical to preparing them for the demands, challenges, and opportunities of the workplace.

Another recommendation to emerge from this study is the use of journal writing to promote and reinforce reflective practice. In addition to the benefits described earlier, the guided journals encouraged students to recognize their accomplishments throughout the project, and reflect on their personal development of important professional skills (i.e., problem solving, collaboration and teamwork, inquiry, and lifelong learning).

A final recommendation is the provision of early and frequent opportunities for students to work on professional problems of practice. The context of this study was the use of PBL in a capstone course. It was not until the end of the academic program for these students that they had an opportunity to apply content learned in their core courses to real projects involving real clients. Nor had they developed or practiced the use of important professional skills such as problem solving, collaboration and teamwork, inquiry, and lifelong learning. According to their journal responses, students were concerned about their ability to be effective in their profession and find meaningful employment, especially in a competitive market suffering from a downturn in the economy. Some students even admitted that they had considered switching majors because they lacked confidence in their ability to succeed in the software engineering industry. These concerns were directly related to their lack of authentic problem-solving experience. One recommendation that seems obvious is the importance of involving students in authentic and real problem-solving activities throughout their academic programs, not just at the end. In addition, although a one-shot experience with real-world problem solving can help, as indicated by this study, frequent exposure to the activities of the community of practice would reinforce, extend, and sustain improvements to self-efficacy, professional identity, and overall performance.

CONCLUSION

This study gave software engineering students a voice by allowing them to describe—in their own words—the changes they were experiencing and the accomplishments they were achieving during a PBL experience. The guided journals show a change in student perceptions about their abilities to be software development professionals—from a lack of self-efficacy prior to their semester-long PBL experience to confidence at the end of the semester. These findings were reinforced by the results of the General Perceived Self-Efficacy Scale, which indicted a significant positive change in student perceptions of personal ability and preparedness for the software development profession. These results are congruent with Bandura's (1977) assertion that the most influential source of self-efficacy information is performance accomplishments.

Based on an analysis of the perceptual changes described in the students' guided journals, it appears that PBL—specifically the instructional strategies of authentic activities, collaboration, and reflection—may help students to experience success, improving their confidence to engage in similar activities in the future, and empowering them to pursue challenges in the field. By engaging students in learning and problem-solving activities that reflect the true nature and requirements of the workplace, PBL may help students feel prepared to work effectively in their profession. □

Joanna C. Dunlap [Joni.Dunlap@cudenver.edu], is at the School of Education of the University of Colorado at Denver and Health Sciences Center.

Special thanks to Dr. Scott Grabinger, University of Colorado at Denver and Health Sciences Center, for his thorough review of this manuscript and for his continuing support as a mentor and colleague.

REFERENCES

Albanese, M., & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. *Academic Medicine, 68*(1), 52-81.

Argyris, C., & Schön, D. (1987). *Theory into practice: Increasing professional effectiveness*. San Francisco: Jossey-Bass.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84*, 191-213.

Bandura, A. (1986). *Social foundations of thought and action: A social-cognitive theory*. Englewood Cliffs, NJ: Prentice-Hall.

Bandura, A., & Schunk, D. (1981). Cultivating competence, self- efficacy, and intrinsic interest through proximal self-motivation. *Journal of Personality and Social Psychology, 41*, 586-598.

Barrows, H. S. (1984). A specific, problem-based, self-directed learning method designed to teach medical problem-solving skills, self-learning skills and enhance knowledge retention and recall. In H. Schmidt & M. deVolder (Eds.), *Tutorials in problem-based learning* (pp. 16-32). Maastricht, Netherlands: Van Gorcum.

Barrows, H. S. (1985). *How to design a problem-based curriculum for the preclinical years*. New York: Springer Publishing Company.

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education, 20*(6), 481-486.

Barrows, H. S. (1992). *The tutorial process*. Springfield, IL: Southern Illinois University School of Medicine.

Barrows, H. S, & Kelson A. M. (1993). *Problem-based learning: A total approach to education*. Monograph. Springfield, IL: Southern Illinois University School of Medicine.

Barrows, H. S., & Tamblyn, R. (1980). *Problem-based learning*. New York: Springer Publishing Company.

Berkson, L. (1993). Problem-based learning: Have the expectations been met? *Academic Medicine, 68*(suppl 10), 79-88.

Betz, N., & Hackett, G. (1981). The relationship of career-related self-efficacy expectations to perceived career options in college women and men. *Journal of Counseling Psychology, 28,* 399-410.

Borasi, R., & Rose, B. (1989). Journal writing and mathematics instruction. *Educational Studies in Mathematics, 20*, 347-365.

Boud, D. (1985). *Problem-based learning in education for the professions*. Sydney, Australia: Higher Education Research and Development Society of Australia.

Boud, D., & Feletti, G. (1991). Introduction. In D. Boud & G. Feletti (Eds.), *The challenge of problem based learning* (pp. 13-22). London: Kogan Page.

Brown, J., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 32-42.

Bruer, J. T. (1993). *Schools for thought: A science of learning in the classroom*. Cambridge, MA: MIT Press.

Collins, A., Brown, J., & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum and Associates.

Colliver J. A. (2000). Effectiveness of problem-based learning curricula: Research and theory. *Academic Medicine, 75*, 259-266.

Connor-Greene, P. (2000). Making connections: Evaluating the effectiveness of journal writing in enhancing student learning. *Teaching of Psychology, 27*(1), 44-46.

Davies, P. (2000). Approaches to evidence-based teaching. *Medical Teacher, 22*(1), 14-21.

Engel C. E. (1991). Not just a method but a way of learning. In D. Boud & G. Feletti (Eds.), *The challenge of problem-based learning* (pp. 22-33). London: Kogan Page.

Gillis, A. (2001). Journal writing in health education. *New Directions for Adult and Continuing Education, 90*, 49-58.

Grabinger, R. S., Dunlap, J. C., & Duffield, J. (1997). Student-centered learning environments in action: Problem-based learning. *Association for Learning Technology Journal (ALT-J), 5*(2), 5-17.

Greeno J. G., Collins A. M., & Resnick L. (1996). Cognition and learning. In R. C. Calfee & D. C. Berliner (Eds.), *Handbook of educational psychology* (pp. 15-46). New York: Simon & Schuster Macmillan.

Hak, T., & Maguire, P. (2000). Group process: The black box of studies on problem-based learning. *Academic Medicine, 75*, 769-772.

Hall, D. T. (1987). Careers and socialization. *Journal of Management, 13*, 301-321.

Hettich, P. (1990). Journal writing: Old fare or nouvelle cuisine? *Teaching of Psychology, 17*, 36-39.

Hmelo, C. E., & Evensen, D. H. (2000). Introduction. Problem-based learning: Gaining insights on learning interactions through multiple methods of inquiry. In D. H. Evensen & C. E. Hmelo (Eds.), *Problem-based learning: A research perspective on learning interactions*. Mahwah, NJ: Lawrence Erlbaum Associates.

Hmelo, C. E., Gotterer, G. S., & Bransford, J. D. (1997). A theory-driven approach to assessing the cognitive effects of PBL. *Instructional Science, 25*, 387-408.

Hodges, H. (1996). Journal writing as a mode of thinking for RN-BSN students: A leveled approach to learning to listen to self and others. *Journal of Nursing Education, 35*, 137-141.

Jerusalem, M., & Schwarzer, R. (1992). Self-efficacy as a resource factor in stress appraisal processes. In R. Schwarzer (Ed.), *Self-efficacy: Thought control of action* (pp. 195-213). Washington, DC: Hemisphere.

Johnson, D., & Johnson, R. (1979). Conflict in the classroom: Controversy and learning. *Review of Educational Research, 49*, 51-69.

Jurdak, M., & Zein, R. (1998). The effect of journal writing on achievement in and attitudes toward mathematics. *School Science and Mathematics, 98*, 8, 412-419.

Kelson, A., & Distlehorst, L. (2000). Groups in problem-based learning (PBL): Essential elements in theory and practice. In D. H. Evensen & C. E. Hmelo (Eds.), *Problem-based learning: A research perspective on learning interactions* (pp. 167-184). Mahwah, NJ: Lawrence Erlbaum Associates.

Kerr, S., VonGlinow, M. A., & Schriesheim, J. (1977). Issues in the study of "professionals" in organizations: The case of scientists and engineers. *Organizational Behavior and Human Performance, 18*, 329-345.

Koschmann, T. D. (2001). Dewey's contribution to a standard of problem-based learning practice. In P. Dillenbourg, A. Eurlings & K. Hakkarainen (Eds.), *European perspectives on computer-supported collaborative learning: Proceedings of Euro-CSCL, 2001* (pp. 355-363). Retrieved December 30, 2003, from http://www.mmi.unimaas.nl/euro-cscl/Papers/90.pdf

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-supported problem-based learning: A principled approach to the use of computers in collaborative learning. In T. D. Koschmann (Ed.), *CSCL: Theory and practice of an emerging paradigm* (pp. 83-124). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lowry, N., & Johnson, D. (1981). Effects of controversy on epistemic curiosity, achievement and attitudes. *Journal of Social Psychology, 115*, 31-43.

Mager, R. (1992). No self-efficacy, no performance. *Training, 29*(4), 32-36.

Manz, C., & Manz, K. (1991). Strategies for facilitating self-directed learning: A process for enhancing human resource development. *Human Resource Development Quarterly, 2*(1), 3-12.

McGowen, K. R., & Hart, L. E. (1990). Still different after all these years: Gender differences in professional identity formation. *Professional Psychology: Research and Practice, 21*, 118-123.

Multon, K., Brown, S., & Lent, R. (1991). Relation of self-efficacy beliefs to academic outcomes: A meta-analytic investigation. *Journal of Counseling Psychology, 38*, 30-38.

Norman, G., & Schmidt, H. (1992). The psychological basis of problem-based learning. *Academic Medicine, 67*(9), 557-565.

Norman, G., & Schmidt, H. (2000). Effectiveness of problem-based learning curricula: theory, practice and paper darts. *Medical Education, 34*, 721-728.

Pajares, F., & Miller, M. (1994). The role of self-efficacy and self-concept beliefs in mathematical problem-solving: A path analysis. *Journal of Educational Psychology, 86*, 193-203.

Pajares, F., & Miller, M. (1997). Mathematics self-efficacy and mathematical problem-solving: Implications of using varying forms of assessment. *Journal of Experimental Education, 65*, 213-228.

Patton, M. (1980). *Qualitative evaluation methods*. Beverly Hills, CA: Sage.

Pintrich, P., & De Groot, E. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology, 82*, 33-40.

Schunk, D. (1982). Effects of children's self-efficacy and achievement. *Journal of Experimental Education, 51*(2), 89-93.

Schunk, D. (1983). Ability versus effort attributional feedback: Differential effects on self-efficacy and achievement. *Journal of Educational Psychology, 75*, 848-856.

Schunk, D. (1984a). Sequential attributional feedback and children's achievement behaviors. *Journal of Educational Psychology, 76*, 1159-1169.

Schunk, D. (1984b). Self-efficacy perspective on achievement behavior. *Educational Psychologist, 19*(1), 48-58.

Schunk, D. (1989a). Self-efficacy and cognitive skill learning. In C. Ames & R. Ames (Eds.), *Research on motivation in education: Vol. 3. Goals and cognitions*. San Diego, CA: Academic Press.

Schunk, D. (1989b). Social cognitive theory and self-regulated learning. In B. Zimmerman & D. Schunk (Eds.), *Self-regulated learning and academic achievement: Progress in cognitive development research* (pp. 83-110). New York: Springer-Verlag.

Schwarzer, R. (1993). *Measurement of perceived self-efficacy. Psychometric scales for cross-cultural research*. Berlin, Germany: Freie Universit_t Berlin.

Schwarzer, R. (1997). General Perceived Self-Efficacy in 14 Cultures. Retrieved February 10, 2000: http://www.yorku.ca/faculty/academic/schwarze/world14.htm

Schwarzer, R., & Jerusalem, M. (1995). Generalized Self-Efficacy scale. In J. Weinman, S._Wright, & M. Johnston (Eds.), *Measures in health psychology: A user's portfolio. Causal and control beliefs* (pp. 35-37). Windsor, UK: nferNelson.

Schwarzer, R., Mueller, J., & Greenglass, E. (1999). Assessment of perceived general self-efficacy on the Internet: Data collection in cyberspace. *Anxiety, Stress, and Coping, 12*, 145-161.

Simons, G. F. (1978). *Keeping your personal journal*. New York: Paulist Press.

Spector, J. M. (2003). Problems with problem-based learning: Comments on model-centered learning and instruction in Seel (2003). *Technology, Instruction, Cognition and Learning, 1*.

Stepien, W., & Gallagher, S. (1993). Problem-based learning: As authentic as it gets. *Educational Leadership, 50*(7), 25-28.

Vernon, D., & Blake, R. (1993). Does problem-based learning work? A meta-analysis of evaluative research. *Academic Medicine, 68*(7), 550-563.

Walton, H., & Matthews, M. (1989). Essentials of problem-based learning. *Medical Education, 23*, 542-558.

Watts, R. (1987). Development of professional identity in Black clinical psychology students. *Professional Psychology: Research and Practice, 18*, 28-35.

Whitehead, A. (1929). *The aims of education and other essays*. New York: The Macmillan Company.

Wilkerson, L., & Gijselaers, W. H. (Eds.) (1996). *Bringing problem-based learning to higher education: Theory and practice.* San Francisco: Jossey-Bass.

Williams, S. M. (1993). Putting case-based learning into context: Examples from legal, business, and medical education. *Journal of the Learning Sciences, 2,* 367-427.

Zimmerman, B., Bandura, A., & Martinez-Pons, M. (1992). Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal, 29*, 663-676.

See Appendixes A and B, overleaf.