Cases on Online and Blended Learning Technologies in Higher Education:
Concepts and Practices

Yukiko Inoue
University of Guam, Guam

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE

Hershey · New York

Director of Editorial Content: Kristin Klinger
Senior Managing Editor: Jamie Snavely
Assistant Managing Editor: Michael Brehm
Publishing Assistant: Sean Woznicki
Typesetter: Michael Brehm, Kurt Smith, Jamie Snavely
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

Library of Congress Cataloging-in-Publication Data

Cases on online and blended learning technologies in higher education : concepts and practices / Yukiko Inoue, editor.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-60566-880-2 (hardcover) -- ISBN 978-1-60566-881-9 (ebook) 1. Education, Higher--Computer-assisted instruction--Case studies. 2. Educational technology--Case studies. 3. Blended learning--Case studies. I. Inoue, Yukiko.
LB2395.7.C417 2010
378.1'734--dc22
2009034981

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Editorial Advisory Board

List of Reviewers

Table of Contents

B: Courses

Detailed Table of Contents

B: Courses

Chapter 11
Virtual Organizing Professional Learning Communities through a Servant–Leader Model of Appreciative Coaching

Kam Hou Vat
University of Macau, Macau

ABSTRACT

This case investigates a set of empowerment concerns in the context of transforming classes of student and teacher learners (considered as department-wide learning units in higher education) into professional learning communities (PLCs). In particular, we are interested in enhancing student learning through designing a collaborative learning environment in support of problem-based learning, based on the concept of virtual organizing the various PLCs distributed throughout a higher educational institute. Of specific interest in our exploration is the generative potential of a servant-leader model of student-centered education in support of the PLCs nurtured by the development practice of appreciative coaching adapted from the established positive change paradigm of appreciative inquiry.

ORGANIZATION BACKGROUND

The department of computer and information science (CIS), as a constituent unit of education under the Faculty of Science and Technology at the author's affiliated university, is installed to offer degree programs in both the undergraduate and graduate levels in software engineering. The department has a current population of about 150 undergraduates and 30 graduate students (mostly part-time). It has to coordinate per academic year, the enactment of about 20 graduate and 40 undergraduate courses. There are currently five laboratories installed for the information technology (IT) education of our students: software engineering laboratory, e-commerce technology laboratory, distributed systems laboratory, computer graphics and multimedia laboratory, and the motion capture laboratory. Besides, there are over two hundred PC's distributed on campus, to offer 24-hour computer service to our students, including Internet access. To help manage course delivery, the university also provides course management systems, such as WebCT (since 1998) and MOODLE (since 2008) to teaching staff for their

DOI: 10.4018/978-1-60566-880-2.ch011

course enactment. Currently, the means of education delivery in our department has largely been didactic; yet, we are quite willing to blend the best of our old values of good teaching through the instructivist approach with the modern-day constructivist way of thinking such as problem-based learning (PBL) (Amador, Miles, & Peters, 2006).We are also interested in the continuing efforts to extend our curriculum and instructional practice over the Internet, through some continually renewed electronic (mostly Web-based) course support, both for the teaching staff and for the students.

SETTING THE STAGE

The following case description recounts the action research experience of some bottom-up course-support initiative sustained by individual staff members from the Department of CIS over the years in reshaping our undergraduate learning landscape through the integration of some ICT-enabled (information and communication technologies) environments to enhance student learning. In particular, this report is based on the experience acquired through the experimentation of a Web-enabled course support environment called REAL (Rich Environment for Active Learning) initiated in 1999, and reactivated in 2008 with a renewed title as REALSpace (Vat, 2009b) to nurture an emergent interest of professional learning community (PLC) (Dufour & Eaker, 1998) to be properly described as follows. It is our lessons learned that if student learning is to improve, staff should be well informed of the PLC potential and develop the capacity to function as PLC. If students are to benefit from the PLC, they must develop a collaborative culture. If students are to develop a collaborative culture, we must overcome the tradition of teacher-centered education (teacher as sage on the stage). If schools are to overcome their tradition of teacher-centered education, teachers must learn to work in collaborative teams (as coaches by the side). If schools are to support effective teamwork to enhance student learning, there must be some technology-enhanced environment to enable learning among teachers and students. And the concept of virtual organizing fits right in to provide the mechanism of a learner-centered appreciative knowledge environment (AKE) to stimulate and facilitate a learning-centered culture of knowledge sharing to enhance student achievements. The impact of a servant-leader model of education (Greenleaf, 1977) should serve as a transformative path to enable the learning cycle of appreciative coaching (AC) (Orem, Binkert, & Clancy, 2007) on the part of teachers to enable students to tap into or rediscover their own sense of wonder about their present and future possibilities.

The Context of PLC

The premise in our discussion of PLC (Dufour, Dufour, & Eaker, 2008) lies in the assumptions of the meaning behind the three words: professional, learning, and community. It is believed that a *professional* is someone with expertise in a specialized field, an individual who has not only pursued advanced training to enter the field, but who is also expected to remain current in its evolving knowledge base. The term *learning* suggests ongoing action and perpetual curiosity. It is expected that if students are to learn, those who educate them must engage in the ongoing study and constant practice of their field. The term *community* suggests a group linked by common interests that provide members with a sense of identity, belonging, and involvement that result in a Web of meaningful relationships with moral overtones (Sergiovanni, 2005, p55). Communities (or communities of practice) form around common characteristics, experiences, practices, or beliefs that are important enough to bind members to one another in a kind of fellowship (Wenger, 1998). Successful communities should provide members with broadly shared opportunities to

participate, promote collective responsibility, and foster a strong sense of belonging (Clinton, 2007). In a PLC, all of the above characteristics are evident. Educators create an environment that fosters shared understanding, a sense of identity, high levels of involvement, mutual cooperation, collective responsibility, emotional support, and a strong sense of belonging as they work together to achieve what they cannot accomplish alone. The essence of the PLC is a focus on and a commitment to the learning of each student (Dufour & Eaker, 1998). To achieve this shared purpose, members of a PLC are expected to create and are guided by a clear and compelling vision of what their school (or teaching) must become to help all students learn. This often requires of the PLC members to make collective commitments that clarify what each member will do to contribute to creating a PLC, and to use results-oriented goals to mark their progress. Still, one of the major challenges in the implementation of the PLC concept is convincing educators to move beyond the question, "Was it taught?" to the far more relevant question, "Was it learned?" This case description advocates for a new culture of learning that addresses how educators will work to improve their teaching, and subsequently student learning.

The Servant-Leader Model of Education

The term servant-leadership was first coined in a 1970 essay by Robert K. Greenleaf entitled *The Servant as Leader* (Greenleaf, 1977; http://www.greenleaf. org). As a lifelong student of how things get done in organizations, Greenleaf distilled his observations in a series of essays and books on the theme of servant leadership – the objective of which was to stimulate thought and action for building a better, more caring society. The idea of the servant as leader came partly out of Greenleaf's half century of experience in working to shape large institutions. The central meaning of servant leadership was interpreted by Greenleaf as follows: The great leader is first experienced as a servant to others, and that this simple fact is central to his or her greatness. True leadership emerges from those whose primary motivation is a deep desire to help others. In all of his works, Greenleaf discusses the need for a better kind of leadership model, a model that puts serving others – including students, employees, customers, and community – as the number one priority. Servant leadership emphasizes increased service to others, a holistic approach to work, promoting a sense of community, and the sharing of power in decision making. In *The Servant as Leader*, Greenleaf wrote (Beazley, Beggs, & Spears, 2003):

It begins with the natural feelings that one wants to serve, to serve first. Then conscious choice brings one to aspire to lead. The difference manifests itself in the care taken by the servant – first to make sure that other people's highest priority needs are being served. The best test is: do those served grow as persons; do they, while being served, become healthier, wiser, freer, more autonomous, more likely themselves to become servants? (p. 16)

Thereby, at its core, servant leadership is a long-term, transformational approach to life and work – in essence, a way of being – that has the potential for creating positive change throughout our society. In the setting of education, the context of servant leadership brings forth the concept of teacher as servant, carrying the connotation that student-centered education serves to shift from a focus on teaching to a focus on student learning.

The Potential of Appreciative Coaching

The practice of AC attributed to (Orem, Binkert, & Clancy, 2007) is developed from the established change management paradigm of appreciative inquiry (AI) whose philosophy is based on the

assumption that inquiry into and dialogue about strengths, successes, hopes and dreams is itself a transformational process (Cooperrider & Whitney, 2005). AC describes an approach to coaching that shows individuals how to tap into or rediscover their own sense of wonder and excitement about their present life and future possibilities. Rather than focusing on individuals in limited or problem-oriented ways, AC is meant to guide individuals through different stages of appreciative development: discovery, dream, design and destiny – that inspire them to an empowering view of themselves and their future. The core process of AC begins with the selection of a topic. In the context of our discussion, the topic chosen for student-centered education has been "enhancing student learning through implementing a PLC of problem-based learning." At the outset of the coaching relationship such as in the discovery stage, core questions serve to explore the client's strengths, past successes, work and personal values, and the one or two things he or she longs to have more of or to have being different in life. From the answers to these questions come the tools for learning and change. In practice, trust should begin to build in the coaching relationship when clients can experience some positive feelings about themselves and their situation. In the dream stage, client and coach come together to make sense of the answers to the core questions so that they may apply these answers to the topic. In this stage, we are using the client's proudest accomplishments, core skills and strengths, and deepest values to create something with which we can explore and experiment. Once the client could bring his or her dream into clear view, it is time to design a plan for the dream. The design stage relates to the ongoing dance between coach and client of defining, performing, and assessing experiments. Design implies a plan or an impression or a mock-up of some future reality. There is no assumption that an initial design is the final design. Experimentation is the order of the day. The ultimate design should incorporate as many of the skills and strengths of the client as is possible or appropriate. Typically, clients step into the destiny stage once they have begun to implement the concrete actions and practices they identified and designed in the design stage for realizing their desired future. The destiny stage is a time for clients to acknowledge and celebrate the accomplishments they are making in either moving toward or actually realizing their dream. At the conclusion of this stage, clients may choose to move to a second cycle of AC by expanding on other elements of their dream or creating a new dream. This is an excellent opportunity for coaches to help client reflect on the work they have done and appreciate the result they have achieved. This AC process of emphasizing the positive seems in most cases to generate positive feelings, increased energy, and a deeper connection to oneself. It is nonetheless true that exploring one's innermost desires should turn out to be a pleasurable experience.

The Flexibility of Virtual Organizing

The idea of virtual organizing, attributed to Venkatraman and Henderson (1998), can be considered as a method of operationalizing a PLC, dynamically assembling and disassembling nodes on a network of people or groups of people, to meet the demands of a particular learning context. This term emerged in response to the concept of virtual organization, which appeared in the literature around the late twentieth century (Byrne, Brandt, & Port 1993; Davidow & Malone 1992; Hedberg, Dahlgren, Hansson, & Olve 1997). There are two main assertions associated with virtual organizing. First, virtual organization should not be considered as a distinct structure such as a network organization in an extreme and far-reaching form (Jagers, Jansen, & Steenbakkers 1998), but virtuality is a strategic characteristic applicable to every organization. Second, IT is a powerful enabler of the critical requirements for effective virtual organizing. In practice, virtual organizing helps emphasize the ongoing process nature

of the organization, and it presents a framework of achieving virtuality in terms of three distinct yet interdependent vectors: virtual encounter for organization-wide interactions, virtual sourcing for asset configuration, and virtual expertise for knowledge leverage. The challenge of virtual organizing is to integrate the three hitherto separate vectors into an interoperable IT platform that supports and shapes the new organizational initiative, paying attention to the internal consistency across the three vectors.

CASE DESCRIPTION

The teaching of *SFTW 300 Software Psychology*, a junior core course more properly renamed as human-computer interactions (HCI) (Vat, 2001) in the undergraduate curriculum of Software Engineering offered by the Department of Computer and Information Science, has always been a challenge as it is composed of such a mix of elements as human factors, user expectations, man-machine interfaces construction, cognitive psychology, computer science, and those latest developments on contextual design in interactive systems (IS). In the case of the author's teaching experience, since 1998, the pedagogy adopted to deliver such a course has been shifted from a conventional instructivist approach to the constructivist method of problem-based learning (PBL) (Greening, 2000). Besides, with the increasingly accumulated course materials to cover in a single semester, the idea of scenario-based design (Carroll, 2000) has also been incorporated starting from 2000 with an attempt to help undergraduate Software Engineering students deepen the idea that HCI is concerned with understanding, designing, evaluating and implementing interactive computer systems to match the needs of people. It is our experience that the constructivist's ideas of PBL (Barrows, 1986) revolving around a focal problem, group work, feedback, skill development and iterative reporting, with the instructor playing the coach by the side, guiding, probing, and supporting student-groups' initiatives along the way, could help students develop a unified team-based approach to better manage the underlying software requirements. Methodically, we still need some working scenarios to try out the iterative learning process involving researchers (instructor) and practitioners (students) acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning. In particular, our action research approach should involve evaluating how well the students playing the role of practitioners, could function as self-directed work teams (SDWTs) of software professionals, following the constructivist's tenets of PBL, in performing group-based software development for specific user scenarios. Against this backdrop, the use of AC has demonstrated quite a promise in enhancing the student-practitioners' learning to deal with the design difficulties typified in the complex domain of ill-defined problem situations.

Recognizing the PBL Potential of Student Collaboration

PBL, (Greening, 2000; Ryan, 1993; Barrows, 1986), as a pedagogy, is designed to actively engage our students, divided in groups, in opportunities for knowledge seeking, for problem solving, and for the collaborating necessary for effective practice. At the heart of PBL are some complex real-world problems used to motivate students to identify and research the issues and principles they need to know to work through those problems (Boud & Feletti, 1997). The design of a PBL-based curriculum addresses directly many of the recommended and desirable outcomes of an undergraduate education; specifically, the ability to do the following (Boyer, 1998):

- think critically and be able to analyze and solve complex, real-world problems;

- find, evaluate, and use appropriate learning resources;
- work cooperatively in teams and small groups;
- demonstrate versatile and effective communication skills, both verbal and written; and
- use content knowledge and intellectual skills acquired at the university to become continual learners.

Since PBL is often designed to enable group-based project work among small teams of students around a set of teamwork activities, including climate setting, starting a problem, following up the problem, and reflecting on the problem, a brief description of the PBL cycle of collaboration is thereby helpful to understand its potential for student collaboration:

The Climate Setting Phase

At the outset, before the PBL group work begins, students must get to know one another, establish ground rules, and help create a comfortable climate for collaborative learning. Meeting in a small group for the first time, students typically introduce themselves, stressing their academic backgrounds to allow the facilitator (instructor) and each other to understand what expertise might potentially be distributed in the group. The most important task is to establish a non-judgmental climate in which students recognize and articulate what they know and what they do not know.

The Problem Initiation Phase

The actual PBL episode begins by presenting a group of students with minimal information about a particular problem. Students then query the given materials to determine what information is available and what they still need to know and to learn to solve the problem. During this phase, students typically take on specific roles. An example is the scribe, who records the group's problem solving, including listing the facts known about the problem, students' ideas, additional questions about the problem, and the learning issues generated throughout ensuing discussion. Such written record helps the students keep track of their problem solving and provides a focus for negotiation and reflection. Throughout the problem-solving process, students are encouraged to pause to reflect on the data collected, generating additional questions about that data, and hypothesizing about the problem and about possible solutions. Early in the PBL process, the facilitator may question students to help them realize what they do not understand. As students become more experienced with the PBL method and take on more of the responsibility for identifying learning issues, the facilitator is able to fade this type of support, or scaffolding. After the group has developed its initial understanding of the problem, the students divide up and independently research the learning issues they have identified. The learning issues define the group's learning goals and help group-members work toward a set of shared objectives. These objectives can also help the facilitator to monitor the group's progress and to remind members when they are getting off course, or alternately, to ask if they need to revise their goals.

The Problem Follow-up Phase

In the problem follow-up phase, students reconvene to share what they have learned, to re-consider their hypotheses, or to generate new hypotheses in light of their new learning. These further analyses, and accompanying ideas about solutions, allow students to apply their newly acquired knowledge to the problem. Students share what they have learned with the group as they interpret the problem through the lens of their newly accessed information. At this point, it is important for the students to evaluate their own information and that of the others in their group. In the PBL

group, information is not often accepted at face value. Students must discuss how they acquired their information and critique their resources. This process is an important means of helping the students become self-directed learners.

The Problem Reflection Phase

During post-problem reflection, students deliberately reflect on the problem to abstract the lessons learned. They consider the connections between the current problem and previous problems, considering how this problem is similar to and different from other problems. This reflection allows them to make generalizations and to understand when this knowledge can be applied. Finally, as the students evaluate their own performance and that of their peers, they reflect on the effectiveness of their self-directed learning and their collaborative problem solving.

Consequently, PBL acknowledges the possibility of prior knowledge held by the learner. Further knowledge is acquired on a need-to-know basis, enabling the learner to diagnose his or her own learning needs. Knowledge gained is fed back into the problem in an iterative loop (Margetson, 1994). PBL allows the synthesis of topics and subjects. According to Woods (1994), one specific advantage of this approach is increased motivation; namely, learners learn because they are interested. More importantly, Woods maintains that because of the way in which knowledge is acquired in PBL, links are provided with experience which help in future recall. This is invaluable for students' future professional life (Barrows, 1986).

Supporting PBL Online

In the context of our PBL learning design elaborated in the following course scenario, the basic online support comes mainly from the MOODLE environment (www.moodle.org) which is an ongoing open-source development project to support a social constructionist framework of education. Simply stated, this style of learning and teaching from MOODLE, short for modular object-oriented dynamic learning environment, is based on four main concepts (http://docs.moodle.org/en/Philosophy):

Constructivism: The constructivist believes that students actively construct new knowledge as they interact with their environments, including their course activities and other students.

Constructionism: The constructionist asserts that students learn more when they construct learning experiences for others. We might be familiar with the learning pyramid (http://homepages.gold.ac.uk/polovina/learnpyramid/index.html) which states that students remember 10% of what they read, 20% of what they hear, 30% of what is demonstrated to them, 50% of what they discuss, and 75% of what they practice. That same pyramid also states that students retain 90% of what they teach others.

Social Constructionism: This extends constructionism into social settings where groups construct knowledge for one another, collaboratively creating a small culture of shared artifacts with shared meanings. When students become part of a culture, they are constantly learning. For example, in the context of ballroom dancing, there is a large difference between watching a video showing people dancing, and practicing in a class with other students and possibly a variety of teachers. The latter would enrich and accelerate our learning process.

Separate and connected behavior: The context of connected and separate behaviors come from the study of human motivations: separate behavior occurs when people try to remain objective and factual, tending to defend their ideas by pinpointing holes in their opponents' ideas. Connected behavior is the empathic approach that accepts subjectivity, trying to listen and ask questions in an effort to understand others' point of view. It is convinced that a healthy amount of connected behavior within a learning community is a very powerful stimulant for learning, because it not

only brings people closer together but promotes deeper reflection and re-examination of their existing beliefs.

Defining the Course Scenario of Student-Centered Education

At each semester when *SFTW 300 Software Psychology* is offered, our course scenario begins when the instructor helps the class evolve into its team-based organization. Typically, students embark on the PBL cycle of learning through organized groups of 4-6 members (one being the team leader). Each PBL group will be given a dual role to explore as client and as developer within a specified period of time. Namely, each team, acting as the developer, is to complete an interactive systems (IS) design and prototype for another team acting as the client. Yet, the same team is the client of another group, responsible for clarifying the project, and resolving ambiguities as they arise, but in any pair of PBL teams (say, A and B), they cannot be the client and the developer of each other at the same time. It should be noted that an even number of teams is desirable to facilitate pair-wise client-developer interaction. Meanwhile, the instructor, more appropriately called the facilitator, acts as project sponsor for each client team, and as project supervisor for each developer team. Each client team is handed a design project by the sponsor. It is then given some inception time to elaborate on the specifics of the project. At the end of the inception period, each client team is assigned a developer team from among the remaining client teams. After a developer team has been identified, the working and performance of the developer team is guided and monitored by the project supervisor played by the instructor. In a typical semester, there might easily be six to ten PBL teams of students, with each team composed of four to six members each.

Essentially, each design project invites our PBL student-groups to embark on a journey to develop some IS that meets customers' real needs in Web-based development. The general requirement is for each PBL team to create and maintain a review Web-site to keep all team members up-to-date on all possible aspects of the project. It is also where the PBL team will work (report) collaboratively on the project. Through the review Web-site, our PBL teams can conduct reviews with their clients, who can view their project in progress, give feedback on a design, get in touch with the developer PBL team, and check the project schedule. The review Web-site contains numerous information such as: the roles and responsibilities of the project team, contact information for all team members, the project mission, the vision document, the project schedule, and all design reviews. It is designed that the first thing our PBL teams have to learn is a systematic approach to eliciting, organizing, and documenting the requirements of the system to be built for the client team. Also important is a process that establishes and maintains continuous agreement between the client and the developer teams (Curtis, Krasner, & Iscoe, 1988) on the changing requirements of the system. Individual PBL teams have to understand users' problems in their culture and their language and to build systems that meet their needs. Practically, the HCI context for the course is designed around four core development processes to be experienced by our PBL student-groups within the semester's duration constraint.

Analyzing the Problem. This involves a set of skills to understand the problem to be solved before application development begins. It is the process of understanding real-world problems and user needs and proposing solutions to meet those needs. We consider a problem as the difference between things as perceived and things as derived (Gause & Weinberg, 1989). Accordingly, if the user perceives something as a problem, it is a real problem, and it is worthy of addressing.

*Understanding User Need*s. Software teams are rarely given effective requirements specifications for the systems they are going to build. Often they have to go out and get the information they

need to be successful. Typical methods include interviewing and questionnaires, requirements workshop, brainstorming and idea reduction, storyboarding, role playing, and prototyping. Each represents a proactive means of pushing knowledge of user needs forward and thereby converting fuzzy requirements to those that are better recognized.

Defining the System. This describes the process by which the team converts an understanding of the problem and the users' needs to the initial definition of a system or application that will address those needs. Our PBL teams should learn that complex systems require adaptive strategies to organize information for requirements. This information could be expressed in terms of a hierarchy, starting with user needs, transitioning through feature sets, then into the more detailed software requirements.

Managing the Project Scope. Project scope is presented as a combination of the functionality to be delivered to meet users' needs, the resources available for the project, and the time allowed in which to achieve the implementation. The purpose of scope management is to establish a high-level requirements baseline for the project. The team has to establish the rough level of effort required for each feature of the baseline, including risk estimation on whether implementing it will cause an adverse impact on the schedule.

Throughout the course delivery, each PBL team is required to present their work in progress, and lead class forums to elicit students' discussions. The team leader, equivalent to project manager, has to coordinate the team activities, and ensure effective team communications. And team members have to help set the project goals, accomplish tasks assigned, meet deadlines, attend team meetings and participate in editing project documents and integrating work-products to be combined as the final project report. At the end of each project milestones, each member of the respective PBL teams is required to make a presentation of his or her project involvement, with a question and answer session for the client team and the whole class. The instructor, acting as the project sponsor for each client team, and as the project supervisor for each developer team, designs the necessary scenario details to guide, motivate and provide feedback to the PBL groups. Also, the instructor has to evaluate how well students perform in the PBL groups and how well such groups behave as SDWTs in managing software requirements (Conklin & Burgess-Yakemovic, 1991), and provide the necessary adjustments for the scenarios. Typically, there are a number of milestones set for project teams throughout the semester. In particular, there will be a milestone for all client teams to present their systems of interest, followed by the milestones for all developer teams to fulfill the system design, prototyping, and final delivery. At the completion of each milestone, each PBL team will be assessed according to their performance, in terms of the necessary deliverables produced, and the presentation made by the whole team. Records of the team's work should also be available from the team's review Web site for evaluation purpose. There will be a group grade and an individual grade for each member of the team. The group grade is the same for all members, but the individual grade is different. The group grade is given by the instructor and by the whole class, except for the group being evaluated. The individual grade is given through peer evaluation among members of the PBL team. Specifically, a peer evaluation form is created by the group, which is used by each member of the team to rate every other member in the same team. The rating is often divided into three aspects: qualitative comments of the member's work throughout the milestone, the ranking of the member in the group including the evaluator-member, using the scale of 1-to-5 (5: highest performance; 1: lowest performance), and the bonus distribution among all the members, of a specific amount, say, how much each member gets allocated out of 1000 dollars of bonus. In the specific instance of client-developer pair, each developer-team should

also be evaluated by the client-team using a more detailed format because of the direct relationship between the two PBL teams.

Tackling the Issues of Team-Based Collaboration

Students engaged in the attempt to build Web-based support for specific user situation, are reminded of the delicate business of creating a conglomeration of various human activity systems. This endeavor requires the effort and commitment on the part of everyone (client and developer) involved, as well as a good imagination in the mind of those charged with directing its implementation (Fisher, 2000; Ginac, 2000). In the instance of a project team charged with the mission of creating IS support for group-based project work, what makes the team work is people's mutual understanding of their own and others' interests and purposes, and the recognition that their interests are somehow bound up in doing something to which they all contribute. In a strict sense, it is in the course of interaction that people's sense of purpose and even their contributions, come to be defined. As collaborators in an IS team, PBL teams face the tremendous challenge of how team members move from being individual spokespeople to a unified, collaborative body. In his book on group decision-making, Kaner (1996) calls the transition from the divergent zone of the individual to the convergent zone of the team member the "groan zone." In a team, even though every member wants to contribute to success and to get the project going, each has a different perspective, a different experience, or a different context to bring to the project. Each person's thinking is divergent, bringing diversity to the process, but not much agreement. Convergence occurs as the group's individual ideas are integrated into a whole solution. This process of integration does not entail compromise (Fisher, Ury, & Patton, 1991), in which every one gives up something and no one is happy with the result, nor does it mean that everyone is in complete agreement. What convergence means is that everyone has participated and will support the final decision. Kaner calls this period between divergence and convergence the groan zone because it is the time during which team members groan and complain. In the divergent zone, most group members voice their opinions to make sure their ideas being heard by the group. In the groan zone, however, an individual digs behind other people's ideas to try to uncover their reasons, assumptions and mental models. Difficult problems and wrenching decisions cause teams to spend time in the groan zone because of the required interchange, sharing, and resolution of ideas, and viewpoints. Likewise, the groan zone is also used to describe the transition zone in which innovative, emergent (or unexpectedly desirable) results are generated. Indeed, collaborative groups, especially those in fast-paced environments, groan a lot. They struggle to create the services that converge on the mission profile. They struggle to integrate their own, and others' diverse perspectives. Rather than focusing on individuals in limited or problem-oriented ways, it is here that AC should guide students through different stages of development, including discovery, dream, design, and destiny – that inspire them to an appreciative and empowering view of themselves and the team's future. By an AC model of student (or team) empowerment, the underlying belief is the experiencing of the situation by beginning with a grounded observation of the best of what is, articulating what might be, ensuring the consent of those in the system to what should be, and collectively experimenting with what can be. This formulation, in terms of enhancing student learning in group-based project work, could be considered as the open source philosophy behind a school's collaborative core with her constituents – teachers and students; namely, the PLC seconded by the teacher-as-servant state of mentality and practice.

Soothing the Growing Pains of Team Buildup

It certainly takes time and discipline to transform a PBL team of student members into a SDWT of professional software practitioners. In the short span of each *SFTW300 Software Psychology* semester of about three and a half months, there are many soft skills a PBL team needs to acquire. The following represents a useful set of selected concerns worthy of attention in the team buildup process:

Process Focus: The average student has little background in actually accomplishing process steps within a project schedule and more often no background in doing so in a team with the added requirements of several milestones of prototyping and an inflexible delivery date, as well as being a client expected to be process-aware. During the early stages of the class, each PBL team is given upfront lectures on typical software development processes, such as the dynamic systems development method process, the open unified process, and the extreme programming process. The students are invited to try out a process of their own for their development scenario and the team coordinator is encouraged to keep the team focused on the selected process during subsequent project milestones. Students' feedbacks often indicate that their experience with the pressure of the delivery schedule, client involvement, and prototypes development, has taught them that the chosen process must be flexible enough to accept change but the balance between consistent application of a process and responsiveness to the client is not easy to maintain and this know-how is not likely to be acquired through mere lectures alone. The client experience during the semester also demonstrates to each team the expected learning a new college graduate in the field of software development should encounter after graduation, and the lessons learned through the project should remain with them far longer than their conventional individual examinations.

Team Dynamics: Students embarking on *SFTW300* have had one semester's PBL style of collaboration in SFTW241 Programming Languages Architecture (I). However, the grouping arrangements of *SFTW300* invite each newly formed PBL group to discover whom they can rely on, capitalizing on individual skill sets, and finding a way to work together. Students often tend to be mistrusting at first (still very comfortable with individual efforts) and become leery of having to rely on others. The beginning lectures then become essential occasions to conduct what will be the first of many activities that should promote positive group interactions. Examples of such activities include: writing a group portfolio expressing the profiles of individual team members in terms of their individual technical expertise; engaging in mental games that require the skillful use of teamwork to complete or that make a point about the distinction between individual and group-centered learning (or working) styles. This understanding acquired becomes instrumental when different roles are being taken by members of the group: one role taken up by one member, or one role shared by two or three members, or roles being taken by members through rotating turns. The idea is to achieve coordination to get the project work done through a suitable mix of individual work, cooperating work (different tasks done by different members so as to integrate the pieces), and collaborating work (same portions of work done jointly by different members).

Planning Concerns: Throughout the semester's work, there are several essential milestones (essential due dates) that have to be met by each PBL group. Yet, the only hard and fast date that must be rigorously met is the final delivery. Deliverables required of each PBL team include a concept of operations document, a design document, a test plan, and the final prototype comprising site architecture, schematics, and navigation guide in the specific context of a collaborative Web project. In the client-developer relationship established by the instructor, each team coordinator is permitted to

make arguments for extension of any deliverable due date (except for final delivery) knowing that each extension granted added additional difficulties later in the course for an on time delivery of other documents. This procedure forced each team to evaluate their scheduling philosophy and to perform an informal risk assessment for the entire project. The policy of assigning the same project (group) grade to each member of the team is seen as a strong motivator for each student to take seriously the project activities. Following the industrial model of shared responsibility (the team fails or the team succeeds) seems to provide a far more memorable learning experience in the context of planning a process and maintaining a schedule. Rigorous discussions have often been observed over issues of planning milestones and still leaving enough time to produce the remaining deliverables with reasonable quality and timeliness.

Establishing the Collaborative Context of Project Support

It has been our experience in conducting *SFTW 300 Software Psychology* that some electronic project-based support such as a Web portal is needed to manage collaborative project development. This portal should lead to a Web-based organizational space for each project, $\mathbf{OS}_{\mathbf{Project}}$, which renders a number of peculiar services to client and developers, in the form of distributed applications customizable to their project-related activities. In a specific project context, there must also be a number of Web-based collaborative spaces, $\mathbf{CS}_{\mathbf{Group}}$, to enable group-based project work to be performed. For example, there is respectively a $\mathbf{CS}_{\mathbf{Group}}$ for each of the client PBL team and of the developer PBL team. Besides, to support the interactions among project members and between the project manager (mostly played by PBL team leader) and specific team members, the provision of a personal electronic space for each of the project members, $\mathbf{PS}_{\mathbf{Participant}}$ ($\mathbf{PS}_{\mathbf{Client}}$, or $\mathbf{PS}_{\mathbf{Developer}}$) is essential to facilitate individual work. The linkages from the project space, to the respective collaborative spaces, to the individual personal spaces, must be closely coordinated to support the Web-based auxiliary processes of collaboration in project development. The challenge is to ensure that the sites should complement the project work by enabling both client and developer teams to interact asynchronously or synchronously through the different customizable services offered. An expression for this project-based electronic support (Vat, 2004a) could be written as follows: $\mathbf{<IS\text{-}Support>}_{\mathbf{Project}}\mathbf{::=}$ $\boldsymbol{OS}_{\mathbf{Project}} + \{\, \boldsymbol{CS}_{\mathbf{Group}} \,\} + \{\, \boldsymbol{PS}_{\mathbf{Participant}} \,\}$, where the braces {} represents the repetition of the element embedded. It is intended that the provision of the project spaces, the collaborative spaces, and the personal spaces in the Web portal for collaborative project work could facilitate the formation of specific sets of IS support for different human activity systems originated from different project scenarios (Vat, 2006a). To elaborate on the design of collaborative IS support we hereby consider the respective project scenarios of planned and unplanned communication events.

The Scenarios of Planned Communication

Planned communication events in project development are scheduled points in time during which participants exchange information on a specific topic or review a work product. Such events are often formalized and structured to maximize the amount of information communicated and to minimize the time participants spend on communication. Typical planned communication events (Bruegge & Dutoit, 2004) include problem presentation, client reviews, project reviews, peer reviews, and status reviews.

Problem Presentation: The focus here is the presentation of the problem statement which describes the problem, the application domain, and the desired functionality of the system. It should

also include some non-functional requirements such as usability and platform specification. The problem statement in general does not include a complete specification of the system. It is meant to be a preliminary requirements activity that establishes common ground between the client and the developer team.

Client Review: The goal of client reviews is for the client to assess the progress of the development and for the developer to confirm or change the requirements of the system. The client review is used to manage expectations on both client and developer sides and to increase the shared understanding among participants. A client review is conducted as a formal presentation during which developers focus on specific functionality with the client. The review is preceded by the release of a work product, such as a specification document, an interface mock-up, or an evaluation prototype. At the outcome of the review, the client provides feedback to the developers. This feedback may consist of a general approval or a request for detailed changes in definition, functionality or schedule.

Project Review: The goals of a project review are for the project manager (PBL team leader) to assess status and for team members to review subsystem interfaces. Project reviews can also encourage the exchange of operational knowledge across teams, such as common problems encountered with tools or the system. A project review is typically conducted as a formal presentation of individual developer teams during which each team presents its subsystem to the management (project sponsor and supervisor) or to other teams that depend on the subsystem. The review is usually preceded by the release of a document describing the aspects of the system under review. At the close of the review, the specific developer team may negotiate changes in the interfaces and changes in schedule.

Peer Review: The goal of peer review is to increase the quality of a work product produced by any designated team member. It is composed of two steps: the walkthrough and the inspection. During the walkthrough, a member of the developer team presents to the other members of the same team, his or her artifact, say, the line-by-line code, or a sequence of user-interface mock-ups. During inspection, the other members challenge any suspicious area and attempt to discover as many issues as possible based on a predefined list of criteria. Communication among participants is artifact-based. The peer review is similar in nature to the project review (typically involving more than one teams), except that they differ in their formality, their limited audience, and their extended duration within a single team of project members.

Status Review: The focus of status review is the tasks distributed among team members. Status reviews are primarily conducted within a team (say, weekly) and aimed to detect deviations from the task plan and to correct them. Status reviews encourage developers to complete pending tasks. The review of task status encourages the discussion of open issues and unanticipated problems, and thus encourages informal communication among team members. Often, solutions to common issues can be shared and operational knowledge disseminated more effectively when discussed within the scope of a team. Increasing the effectiveness of status reviews normally has a global impact on the team performance.

The Scenarios of Unplanned Communication

In an ideal project, all communication takes place during planned communication events. In practice, it is difficult to anticipate all information needs and plan all communications. In general, issues resulting from a combination of seemingly isolated facts from different areas of the project are difficult to anticipate because no participants could have a global overview of all the facts. Consequently, a project should be prepared to deal with unexpected situations, often under pressure.

We call the communication resulting from such crises unplanned communication events, including requests for clarification, requests for changes, and issue resolution.

Request for Clarification: This request represents the bulk of the communication among developers, clients, and users. Such requests are largely unplanned. A participant may request for clarification about any aspect of the system that seems ambiguous. These requests may occur during informal meetings, e-mails, or any other communication mechanism available to the project. It is worthy to note that if most information needs are handled through requests for clarification, such situations represent symptoms of a defective communication infrastructure. And the result could lead to serious failures downstream owing to misunderstandings and missing and misplaced information.

Request for Change: This request represents a channel to report any problems with the system itself, including its documentation, the development process, or the project organization. Typically, a change request contains such details as the classification (say, severe, moderate, or annoying), a description of the problem (say, rationale, or communication), a description of the desired change (say, a proposed solution). Requests for change are often formalized when the number of participants and the system size is substantial.

Issue Resolution: An issue represents a concrete problem, such as a requirement, a design, or a management problem. To each decision in project work represents an issue to be solved so that development can proceed. Alternatives are possible solutions that could address the issue under consideration. Criteria are desirable qualities that the selected solution should satisfy. During requirements analysis, criteria include nonfunctional requirements and constraints such as usability. During system design, criteria include design goals such as reliability. During project management, criteria include management goals and tradeoffs such as timely delivery versus quality. A decision is the resolution of an issue representing the selected alternative according to the criteria that were used for evaluation and the justification of the selection. Typically, it is only after much discussion (or argumentation) that a consensus is reached or a decision imposed, covering all aspects of the decision process, which includes criteria, justification, explored alternatives, and tradeoffs. Using these issue-modeling concepts in the context of capturing project rationales in review meetings, we often write an agenda in terms of issues that we need to discuss and resolve. We state the objective of the meeting (formal or informal) to be a resolution on these issues and any related sub-issues that are raised in the discussion. We structure the meeting minutes in terms of proposals that we explore during the meeting, criteria that we agree on, and arguments we use to support or oppose proposals. We capture decisions as resolutions and action items that implement resolutions. During subsequent meetings, we review status in terms of the action items that we produced in the previous meetings.

CURRENT CHALLENGES FACING OUR WORK

From the discussion built up so far, it is not difficult to foresee that PBL is the kind of group-based project work which has many educational and social benefits, in particular providing students with opportunities for active learning. However, teaching, directing and managing such project work is not an easy process. This is because projects are often: expensive demanding considerable supervision and technical resources; and complex combining design, human communication, HCI, and technology to satisfy objectives ranging from consolidation of technical skills through provoking insight into organizational practice, teamwork and professional issues, to inculcating academic discipline and presentation skills. In preparing our students to get started, familiar and comfortable

with group-based project work, we need some sort of course support which goes beyond what typical course management systems such as WebCT or MOODLE could accommodate currently. Indeed, the specific characteristics required of such a blended learning environment must be carefully delineated and thoughtfully designed with a practical continual learning scenario in order to stimulate any learner-centered involvements. This section discusses the challenges behind providing such course support, through describing an effort in virtual organizing an AKE in support of the PLC comprising our PBL students.

Devising the Appreciative Knowledge Environment (AKE)

Our major focus of student-centered AC lies in the installation of an appreciative knowledge environment (AKE) in which electronic support for AC to enable collaborative knowledge work among students and between teachers and students is made available, especially in their respective work and study settings. In particular, we can consider the AKE as the creative units for knowledge sharing for people on a number of scales. The smallest is perhaps the environment surrounding one individual trying to solve a problem in his or her course assignments, or a small team or work group, collaborating to find creative solutions in its search for innovations, such as a PBL team seeking innovative ways to satisfy client requirements. Nonetheless, if the motivation behind our AKE were to encourage student responsibility, to make learning meaningful, and to encourage active knowledge construction in the specific curricula of students' study, the naturalistic creation of virtual communities of student-learners in the process of using the underlying AKE services, must be well supported. As a knowledge-support environment, there are many possibilities for services identification. Currently, the challenges of how to enhance the value of course-specific knowledge work have rendered, at least, three main design reflections: 1) support the actual practices and daily tasks of the participants (teachers and students); 2) collect experiences and represent them in an accessible and equitable manner; and 3) provide a framework to guide the knowledge process.

Support the Actual Practices and Daily Tasks of the Participants

The AKE environment should support the actual practices and daily tasks of teachers by helping them guide students' learning process through the creation of a visible history of student work. For students, the AKE should support learning practices and tasks by making the thinking of their peers more visible and by illustrating the process of collaborative problem solving through both individual and group inquiry activities (say, through the installation of various Wiki-based applications). Moreover, from a knowledge integration perspective, the practice of teaching and learning involves developing a repertoire of models for explaining situations (say, in the form of various podcasts). What type of knowledge integration framework can best help students and teachers in their daily practice?

Collect Experiences and Represent them in an Accessible and Equitable Manner

The AKE environment should collect experiences and represent them in an accessible and equitable manner to promote the process of connecting ideas so that participants (students and teachers) can use them in subsequent tasks such as during follow-up clarification and illustration. Communities, if viewed as a network of relationships and resources, can be structured to elicit ideas, develop shared understanding, and promote the integration of a diverse set of perspectives. It is important to investigate the potential of structuring discussions in different ways based on the type of discussion and the associated pedagogical goals.

Linking different types of pedagogical goals to design strategies is a challenging task because most of the students are yet to get accustomed to reflecting on the nature of their contributions.

Provide a Framework to Guide the Knowledge Process

The AKE environment should encourage participants to make sense of their learning by creating a culture where people ask each other for justification and clarification. It is essential to investigate how participants adjust their learning behavior as their peers prompt them to support their ideas with evidence. One strategy is to create some commonly agreed upon criteria and to examine how these criteria are adopted and transformed by community members (mostly students) as they interact with one another. For communities to maintain coherence and develop a sense of what is desirable behavior, it is important that a strong community culture be established with a common set of values and criteria for making contributions. Student communities need a general framework to help define the mission and vision for their knowledge process.

Meeting the Virtual Organizing Challenge for AKE

In order for knowledge sharing within an organization (*SFTW300* PBL teams of students) to be successful, it is convinced that the people involved must be excited about the process of sharing knowledge. Thereby, an appreciative sharing of knowledge must be viewed as the non-threatening and accepting approach that makes people realize what they do can make a difference. One common example is the communities of practice (COP) (Wenger, 1998) (be it physical or online) mentioned earlier in the context of PLC. Many organizations today are comprised of networks of interconnected COPs through which knowledge is created, shared, organized, revised, and passed on within and among these communities. In a deep sense, it is by these communities that knowledge is owned in practice. In anticipation of our students' knowledge challenge in a real-world organization, it is useful to conceive of an AKE based on the context of virtual organizing described earlier, and experiment with how the ideas of its three vectors can be applied to nurture online the growth of different COPs (Wenger, 1998) (or, in our case, more properly called communities of learning (COL) scattered throughout an organization.

Virtual Encountering the AKE

From a nurturing perspective, it is important to identify what COLs are desirable in the organization, and how, if they already exist, but are not already online, to enable them to be online in order to provide more chances of virtual encounter of such communities, to the organizational members. For those communities already online, it is also important to design opportunities of interaction among different online communities, to activate their knowledge sharing. Since it is an important COP practice not to reduce knowledge to an object, what counts as knowledge is often produced through a process of communal involvement, which includes all the controversies, debate and accommodations. This collective character of knowledge construction is best supported online with individuals given suitable IS support to participate and contribute their own ideas. An IS subsystem, operated through virtual encounter, must help achieve many of the primary tasks of a community of learning, such as encouraging student participation, establishing a common baseline of knowledge and standardizing what is well understood so that people in the community can exercise their creative energies on the learning issues of interest to the community's collective growth.

Virtual Sourcing the AKE

From the discussion built up in the first vector, it is not difficult to visualize the importance of identifying the specific expertise of each potential COL in the organization (in our case, the Department of Computer & Information Science), and if not yet available, planning for its acquisition through a purposeful nurture of expertise in various COLs related to different curricula of studies. This vector focuses on creating and deploying intellectual and intangible assets for the organization in the form of a continuous reconfiguration of critical capabilities assembled through different relationships in the network of COLs distributed within and across the department(s). The mission is to set up a resource network, in which the COL is part of a vibrant, dynamic network of complementary capabilities. The strategic leadership challenge is to orchestrate an organization's position in a dynamic network where the organization can carefully analyze her relative dependence on other players in the resource coalition and ensure her unique capabilities.

Virtual Expertising the AKE

It is important to understand that not everything we know can be codified as documents and tools. Sharing tacit knowledge requires interaction and informal learning processes such as storytelling, conversation, coaching, and apprenticeship. The tacit aspects of knowledge often consist of embodied expertise – a deep understanding of complex, interdependent elements that enable dynamic responses to context-specific problems. This type of knowledge is very difficult to replicate. In order to leverage such knowledge, an IS subsystem, operated through virtual expertise, must help hooking people with related expertise into various networks of COLs, in order to facilitate stewarding such knowledge to the rest of the organization.

Preparing Students' Blended Learning in PBL Online

Today, many educational institutions across the world have implemented electronic learning (e-learning) environments, implying that learners and teachers increasingly are integrating both physical and electronic resources, tools and environments within mainstream educational settings. Still, these new environments are yet to have a major impact on learning. This is partly because the blending of real and virtual domains in teaching and learning is challenging for most teachers; yet, it is becoming an essential skill for effective teaching. On the one hand, this new way of teaching and learning has the potential to extend learning methodologies, to open up opportunities for flexible online learning as well as to challenge more traditional methods of course delivery (Vat, 2009a). At the same time, it adds a degree of complexity to educational development and curriculum design. It is experienced that the key to success is to concentrate on not merely thinking of how to integrate different sorts of content resources, but also on developing educational processes that blend online with face-to-face interactions. In this regard, the idea to support PBL online is to empower students to learn through various Web-based materials including text, simulations, video demonstrations, and such resources as chat rooms, message boards, wikis, podcasts, and RSS feeds that have been purposely built for problem-based learning. Indeed, the increasing adoption of PBL and the growth in online support has reflected the current shift away from teaching as a means of transmitting information towards enabling learning as a student-generated activity. PBL online is a timely example of a blended e-learning experience for both teachers and students. In fact, the context of blended e-learning (Eklund, Kay, & Lynch, 2003) offers the possibility of changing our attitudes not only as to where and when learning takes place, but in terms of what resources and tools can sup-

port learning and the ways in which these might be used. In particular, blended e-learning fosters integration of different spaces, allowing students to learn from university, or from home or on the move. It can offer flexibility in the time when learners can participate in courses, reducing or removing restrictions arising from the balancing of school or home commitments with study. It opens up the range of media resources that can be used for learning. The blend of space, time and media offers new possibilities as to the sorts of activities students can carry out and the ways they can collaborate, using available electronic tools. Literally, the integration of physical and online spaces means that communities can form and interact in ways that were previously unimagined. It introduces the possibility of interacting in real time (synchronously) in conjunction with opportunities to collaborate over a period of time (asynchronously). This in turn allows exploration of different forms of dialogue and new types of learning. New media resources and tools open up possibilities for students to create their own resource banks, integrating self-generated intellectual assets with more formal materials sourced from libraries around the world. This brings into question some of the traditional values of education, such as who owns, creates and controls resources and knowledge. New types of learning activities thereby challenge our thinking as to how learning might be facilitated, creating new etiquettes of learning and teaching, and shifting the locus of control from the teacher to the learner.

Lessons Learned for a Teacher-as-Servant

On integrating the essential context of servant-leadership in support of implementing a PLC through virtual organizing the vehicle of AKE for classes of *SFTW300 Software Psychology* students over the past years, it is the author's experience that there are three basic empowerment practices (Vat, 2004b) which should preferably be performed by the teacher-as-servant in order to apply AC to facilitate the PBL style of student collaboration. Meanwhile, the use of electronic portfolio (e-portfolio) (Vat, 2009a) as an ongoing formative assessment tool to keep track of student learning is also found to be very promising.

The Three Practices of Student Empowerment

Enable Students to Determine what they need to Learn through Questioning and Goal Setting

It is convinced that students should work to identify their knowledge and skill deficits, and to develop strategies in the form of personal learning goals for meeting those deficits. Also, they should learn to relate what they know to what they do not know and ask questions to guide their quest for new knowledge. The emphasis is to foster a sense of students' ownership in the learning process. If teachers, through the AKE, can guide the students in the identification of what they already know and what they need to learn, then knowledge gaps and mistakes can be viewed in a positive way such as another opportunity to learn. And students can assume more responsibility in addressing their own learning needs during any instructional unit.

Enable Students to Manage their own Learning Activities

It is convinced that students should be enabled to develop their learning plans, which should describe priorities, instructional tactics, resources, deadlines, roles in collaborative learning situations, and proposed learning outcomes, including presentation and dissemination of new knowledge and skills, if applicable. Traditionally, these instructional events are arranged by teachers to be followed by students throughout a semester or school year, in order to accomplish a specified set

of pre-determined learning or assessment objectives. Yet, in that case, it is not advantageous for students to learn to take the initiative. To manage their own learning activities, students must be guided and supported by the teacher in the AKE, slowly taking on more and more responsibility of their own learning.

Enable Students to Contribute to each other's Learning through Collaborative Activities

It is convinced that students should be motivated and supported in discussing and sharing information. Particularly, we should enable students to become co-builders of the course- or subject-related resources through evaluating and refining the entries their peers put into the AKE. Collaborative learning is quite appealing to achieve that purpose; however, it involves not just creating a group and then dividing up the work. Students must be educated to recognize what they are trying to learn in teamwork, value it, and wish to share that value with others. Teachers must provide this sense of accountability by structuring the group work to include both individual and group assessments.

The Educational Potential of Student e-Portfolios

In order to support the frequent formative assessments of students' work in the PLC culture of learning, it is important not to ignore the educational potential of student e-portfolios as a tool for learning in the PLC. In fact, different portfolios (Stefani, Mason, & Pegler, 2007; Bangert, 2004) have been used by students at traditional universities and colleges where face-to-face teaching is the dominant mode of instructions. For example, course portfolios are those assembled by students for individual courses. They document and reflect upon the ways in which the student has met the outcomes for that particular course. Instructor's endorsement is often required to authenticate the course portfolios. Program portfolios are developed by students to document the work they have completed, the skills they have learned, and the outcomes they have met in an academic department or program. The mentor or appraiser could add comments. It could be a requirement for graduation. Oftentimes, students might use a selection from their program portfolio to show to prospective employers. Whatever the primary focus of engagement with students, the use of e-portfolios inevitably adds a strong online element to the teaching and learning. Institutions need to provide electronic support and services; teachers need access and skills to integrate the e-portfolio application into their overall course design, and students need a wide range of electronic abilities in order to develop their e-portfolios. The underlying pedagogy of e-portfolio use is considered the most significant link with student learning. Our experience has indicated that constructivism (Vat, 2009a) does seem to be the approach worthy of repeated experimentation. The aim of constructivist principles as applied to e-portfolio is to engender independent, self-reliant learners who have the confidence and skill to use a range of strategies to construct their own knowledge (Stacey, 1998; Slavin, 1994). Where students are required to develop and maintain an e-portfolio, they are usually expected to reflect on their learning, consider how to give evidence of their learning and possibly even develop a plan (or a learning contract) of what they would like to learn. In other words, an e-portfolio implementation of constructivism usually implies a considerable level of learner autonomy and initiative, of learner responsibility for their learning and of opportunities to refine their learning based on feedback from the teacher and their peers. More importantly, e-portfolio use can be the basis for several student-centered initiatives (Batson, 2005), including: creating a system of tracking student work over time, in a single course, with students and faculty reflecting on it; having a more fully informed and constantly

updated view of student progress in a program, which is very helpful in formative assessment; aggregating other students' work in a particular course to see how the students as a whole are progressing toward learning goals; and assessing other courses in similar ways that are all part of one major and thus assessing the entire program of study.

Remarks for Continuing Challenge

The software engineering workplace of this century requires professionals who not only have an extensive store of knowledge, but also know how to keep that knowledge up-to-date, apply it to solve problems, and function as part of a team. This view of the software industry compels educators to rethink and reinvent the ways in which software practitioners are prepared. In particular, schooling must extend beyond the traditional preparatory goal of establishing a knowledge base. Schooling must actively engage our students in opportunities for knowledge seeking, for problem solving, and for the collaborating necessary for effective practice. To realize such experiences, educators have looked to the potential of PLC to shift the focus on teaching to a focus on learning, seconded by the constructivist pedagogical designs that are based on the assumption that learning is a product of both cognitive and social interactions in problem-centered environments (Greeno, Collins, & Resnick, 1996; Savery & Duffy, 1994). The adoption of PBL in *SFTW300* is an example of such a design (Vat, 2006b), and the support of PBL online is always a challenge of innovation in learning design because the interrelationship of technology and pedagogy always renders such question as what it means to be a problem-based learner in an online setting. In fact, there has been much criticism in recent years about blended learning environments that fail to create effective settings for learning (Oliver & Herrington, 2003). One plausible reason is due to the peculiar focus on technological rather than pedagogical design. It is suggested that there is a need to revive the concept of learning design rather than to do a simplistic repackaging of the course content into blended learning formats (Mason, 1998). The idea of blending different sorts of media and learning tasks is not a new concept; however, blending e-learning has an added degree of complexity because e-learning allows the blending of different spaces. For example, we can use electronic learning environments within physical teaching spaces; we can work across time zones in real time or asynchronously. For effective blending, we need to have a clear idea of what we want to blend and what we might blend. To understand how to design engaging tasks within blended e-learning, we have to have some knowledge of why we might design specific learning activities in particular ways. Why is one sort of activity appropriate for one learning situation, but not so effective in another? What sort of student activity are we expecting to see online and how might this differ from what we are accustomed to? Once teachers have decided what sort of learning activities they require, they face a further problem.This is the question of how to plan so that there is integration of these activities with appropriate resources, electronic tools and environments, using a range of teaching methodologies. Thinking through all the possible combinations and solutions is complex and demanding. It is experienced that students are motivated by solving authentic problems based on real-world activities that may be carried out non-sequentially and iteratively. Such problems contrast with the sequential orchestration of tasks frequently planned in teacher-centered education. Planning non-sequential activities is more complex and may involve integrating a variety of media and electronic tools across real and virtual spaces. More importantly, blended e-learning brings with it a new order. With effective learning design, the locus of control should shift from the teacher to the learner. This shift, combined with the free flow of information, requires the development of new sorts of relationships and

trust. Ethical issues are of primary importance, and institutions may need to develop or revise strategies to reflect this fact.

CONCLUSION

To conclude this case description, I hereby render some of my perspectives behind adopting PBL in the teaching of *SFTW300*. The educational literature warns against compartmentalized units of study that produce students who cannot integrate the different parts of their knowledge. Although a fully integrated degree was beyond the scope of discussion, many of our conventional (teacher-centered) courses had compartments that bore out the literature's predictions. In effect, any new course designed in the Software Engineering program must be as integrated as possible, if we want our students to bring all their knowledge to bear on solving real-world problems in software development. In this regard, the nurture of independence and collaboration becomes important. Indeed, our conventional courses have been widely criticized for stifling students' initiative in learning. Yet, through PBL, we offer an approach to learning where curricula are designed with problem scenarios central to student learning in each curricular component. Students working in small teams examine a problem situation and, through this exploration, are expected to locate the gaps in their own knowledge and skills in order to decide what information they need to acquire in order to resolve or manage the situation. Lectures, seminars, workshops or laboratories support the inquiry process rather than transmit subject-based knowledge. The starting point should be a set of problem scenarios regardless of whether a module or a whole program is being designed. The scenarios enable students to become independent inquirers and help them to see learning and knowledge as flexible entities. Students should discover they can learn by themselves, using a range of resources. They are aided in learning to do this by the PBL cycle of collaboration, which develop in them their social and mega-cognitive skills. Consequently, students' critical thinking and problem-solving abilities are sharpened. These are crucial to effective project (software) development, especially at the higher levels of analysis and design.

In the specific case of *SFTW300*, students have to go through the process of understanding, designing, implementing and evaluating interactive computer systems to match the needs of client. This is a teamwork development exercise requiring students to work in groups. This is important to prepare their future careers; nonetheless, fresh graduates today are expected to have the ability and experience to perform effectively in group-based project work. *SFTW300* supports groups by identifying specific roles for group members, providing class time and guidelines on group management, monitoring group planning and progress, and conducting formative assessments for group management and reflection on group processes. Students working in a group naturally learn to communicate with one another, which is another goal highly valued by the software industry. In particular, at the end of each problem, PBL students need to turn in a report and to give a presentation, during which each student must demonstrate his or her work with justification, followed by a session on question and answer. All these require the students to have good command of communications skills. Overall, PBL fosters in our PLC of *SFTW300* students such generic skills as group work, planning, problem-solving, independent learning, research skills, writing, and oral presentation. These are university goals and also highly valued by employers in the computing industry. What is needed to support the continual nurture of such PLCs is to realize the generative potential of the engine of servant-leadership in sustaining the concomitant application of appreciative coaching (AC) to help our students to tap into or rediscover their own sense of wonder about their future possibilities.

REFERENCES

Amador, J. A., Miles, L., & Peters, C. B. (2006). *The practice of problem-based learning: A guide to implementing PBL in the college classroom*. Bolton, MA: Anker Publishing.

Bangert, A. (2004). The seven principles of good practice: A framework for evaluating online teaching. *The Internet and Higher Education*, *7*(3), 217–232. doi:10.1016/j.iheduc.2004.06.003

Barrows, H. S. (1986). A taxonomy of problem-based learning methods. *Medical Education*, *20*(6), 481–486. doi:10.1111/j.1365-2923.1986.tb01386.x

Batson, T. (2005). *The electronic portfolio boom: What's it all about*? Retrieved March 10, 2009, from http://www.campustechnology.com/article.asp?id=6984

Beazley, H., Beggs, J., & Spears, L. C. (Eds.). (2003). *The servant-leader within: A transformative path*. New York: Paulist Press.

Boud, D., & Feletti, G. (1997). *The challenge of problem-based learning*. London: Kogan Page.

Boyer, E. L. (1998). *Boyer commission on educating undergraduates in the research university for the Carnegie Foundation for the Advancement of Teaching*. Retrieved December 9, 2008, from http://naples.cc.sunysb.edu/Pres/boyer.nsf

Bruegge, B., & Dutoit, A. H. (2004). *Obejct-oriented software engineering using UML, patterns, and Java*. Upper Saddle River, NJ: Prentice Hall.

Byrne, J. A., Brandt, R., & Port, O. (1993, February 8). The virtual corporation. *Business Week*, 36–41.

Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. Cambridge, MA: MIT Press.

Clinton, W. (2007, May 7). *Middlebury College graduation address*. Retrieved December 8, 2008, from http://www.middlebury.edu/about/newsevents/ archive/2007/newsevents_633158738134202567.htm

Conklin, J., & Burgess-Yakemovic, K. C. (1991). A process-oriented approach to design rationale. *Human-Computer Interaction*, *6*, 357–391. doi:10.1207/s15327051hci0603&4_6

Cooperrider, D. L., & Whitney, D. (2005). *Appreciative inquiry: A positive revolution in change*. San Francisco, CA: Berrett-Koehler.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*(11), 1268–1287. doi:10.1145/50087.50089

Davidow, W. H., & Malone, M. S. (1992). *The virtual corporation – structuring and revitalizing the corporation for the 21st century*. New York: HarperCollins.

Dufour, R., Dufour, R., & Eaker, R. (2008). *Revisiting professional learning communities at work: New insights for improving schools*. Bloomington, IN: Solution Tree.

Dufour, R., & Eaker, R. (1998). *Professional learning communities at work: Best practices for enhancing student achievement.* Bloomington, IN: Solution Tree.

Eklund, J., Kay, M., & Lynch, H. (2003). *E-learning: Emerging issues and key trends*. Retrieved March 3, 2009, from http://knowledgetree.flexiblelearning.net.au/edition04/pdf/critique_eklund_kay.pdf

Fisher, K. (2000). *Leading self-directed work teams: A guide to developing new team leadership skills*. New York: McGraw Hill.

Fisher, R., Ury, W., & Patton, B. (1991). *Getting to yes: Negotiating agreement without giving in*. UK: Penguin Books.

Gause, D., & Weinberg, G. (1989). *Exploring requirements: Quality before design*. New York: Dorset House Publishing.

Ginac, F. O. (2000). *Creating high performance software development teams.* Upper Saddle River, NJ: Prentice Hall.

Greening, T. (2000). Emerging constructivist forces in computer science education: Shaping a new future? In T. Greening (Ed.), *Computer science education in the 21st century* (pp. 47-80). New York: Springer.

Greenleaf, R. K. (1977). *Servant leadership: A journey into the nature of legitimate power and greatness.* New York: Paulist Press.

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

Hedberg, B., Dahlgren, G., Hansson, J., & Olve, N. (1997). *Virtual organizations and beyond: Discover imaginary systems*. New York: John Wiley & Sons.

Jagers, H., Jansen, W., & Steenbakkers, W. (1998). *Characteristics of virtual organizations*. Retrieved March 3, 2009, from http://primavera.fee.uva.nl/PDFdocs/98-02.pdf

Kaner, S. (1996). *Facilitator's guide to participatory decision-making.* Philadelphia, PA: New Society Publishers.

Margetson, D. (1994). Current educational reform and the significance of problem-based learning. *Studies in Higher Education*, *19*, 5–19. doi:10.1080/03075079412331382103

Mason, R. (1998). Models of online courses. *ALN Magazine, 2*(2). Retrieved February 4, 2009, from http://www.aln.org/publications/magazine/v2n2/mason.asp

Oliver, R., & Herrington, J. (2003). Exploring technology-mediated learning from a pedagogical perspective. *Journal of Interactive Learning Environments*, *11*(2), 111–126. doi:10.1076/ilee.11.2.111.14136

Orem, S. L., Binkert, J., & Clancy, A. L. (2007). *Appreciative coaching: A positive process for change*. San Francisco, CA: Jossey-Bass.

Ryan, G. (1993). Student perceptions about self-directed learning in a professional course implementing problem-based learning. *Studies in Higher Education*, *18*, 53–63. doi:10.1080/03075079312331382458

Savery, J. R., & Duffy, T. M. (1994). Problem-based learning: An instructional model and its constructivist framework. *Educational Technology*, *35*(5), 31–38.

Sergiovanni, T. (2005). Strengthening the heartbeat: Leading and learning together *in schools.* San Francisco, CA: Jossey-Bass.

Slavin, R. (1994). Student teams and achievement divisions. In S. Sharon (Ed.), *Handbook of cooperative learning methods* (pp. 3-19). Westport, CT: Greenwoeed Press.

Stacey, E. (1998). Learning collaboratively in a CMC environment. In G. Davies (Ed.), *Teleteaching 98: Distance learning, training, and education* (pp. 951-960). Proceedings of the XV IFIP World Computer Congres, Vienna and Budapest.

Stefani, L., Mason, R., & Pegler, C. (2007). *The educational potential of e-portfolios: Supporting personal development and reflective learning*. New York: Routledge.

Vat, K. H. (2001, June 25-27). Teaching HCI with scenario-based design: The constructivist's synthesis. Proceedings of the *Sixth Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 9-12*)*, Canterbury, UK.

Vat, K. H. (2004a, February 27-28). Towards a learning organization model for PBL: A virtual organizing scenario of knowledge synthesis. In *CD-Proceedings of the Seventh Annual Conference of the Southern Association for Information Systems*, Savannah, Georgia, USA.

Vat, K. H. (2004b, December 15-17). Toward a learning organization model for student empowerment: A teacher-designer's experience as a coach by the side. *Proceedings of the 2004 IADIS International Conference on Cognition and Exploratory Learning in Digital Age* (pp.131-140), Lisbon, Portugal.

Vat, K. H. (2006a). Conceiving scenario-based IS support for knowledge synthesis: The organization architect's design challenge in systems thinking. *Journal of Systemics, Cybernetics and Informatics, 3*(3). Retrieved March 3, 2009, from http://www.iiisci.org/Journal/CV$/sci/pdfs/P222366.pdf

Vat, K. H. (2006b). Developing a learning organization model for problem-based learning: The emergent lesson of education from the IT trenches. *Journal of Cases on Information Technology*, *8*(2), 82–109.

Vat, K. H. (2009a). An e-portfolio scheme of flexible online learning. In P. L. Rogers, G. A. Berg, J. Boettcher, C. Howard, L. Justice, & K. Schenk (Eds.), *Encyclopedia of distance learning* (pp. 941-949). Hershey, PA: Information Science Reference.

Vat, K. H. (2009b). Developing REALSpace: Discourse on a student-centered creative knowledge environment for virtual communities of learning. *International Journal of Virtual Communities and Social Networking*, *1*(1), 43–74.

Venkatraman, N., & Henderson, J. C. (1998). Real strategies for virtual organizing. *Sloan Management Review*, *40*(1), 33–48.

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. New, York: Cambridge University Press.

Woods, D. (1994). *How to gain the most from problem-based learning*. McMaster, Canada: McMaster University.