DOCUMENT RESUME

ED 405 830 IR 018 270

AUTHOR	Smith, Peter D.
TITLE	A Process Education Approach To Teaching Computer Science.
PUB DATE	96
NOTE	11p.; In: Association of Small Computer Users in Education (ASCUE) Summer Conference Proceedings (29th, North Myrtle Beach, SC, June 9-13, 1996); see IR 018 247.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Active Learning; *Computer Science Education; *Cooperative Learning; Educational Change; Higher Education; Instructional Development; *Journal Writing; Laboratories; *Learner Controlled Instruction; Microcomputers; *Process Education; Student Reaction; Teacher Role; Teamwork

ABSTRACT

The driving force of process education is its focus on students' "learning to learn." This paper describes an approach to teaching computer science which includes classroom management; the adaptation of four different courses to follow the process education approach; successes achieved; and students' responses. The courses are conducted in closed labs, a shift from lecture-centered to lab-centered teaching. Team projects are assigned and each member of each four-person team is given a defined role to assume: team captain, recorder, reflector and spokesperson. At the end of each class, the team is responsible for producing a report of the work products for the class period. In addition, students keep journals with one entry for each class. At the end of the first two weeks, students choose the weight they wish to assign to each aspect of the course. By the fourth week, teams are cooperating fully, and students are convinced they are learning better. Journals are collected at least four times a semester and grades are assigned on how conscientiously they were completed. A mid-course assessment is also scheduled where teams are asked to brainstorm and come up with better ways to organize the class. Each of the four computer science courses that were adapted to use process education is project-oriented and to a certain extent a lab-centered course. With process education, the role of the teacher changed from that of lecturer to one taking on the roles of leader, assessor, facilitator, and evaluator. An appendix describes the team roles and lists performance criteria for each role. (Contains eight references.) (AEF)

A PROCESS EDUCATION APPROACH TO TEACHING COMPUTER SCIENCE

Peter D. Smith
Mathematics Department
Saint Mary's College
Notre Dame, IN 46556
(219) 284-4493
psmith@saintmarys.edu

Introduction

Throughout this paper, I will be discussing "Process Education." The collection of concepts, techniques, and way of thinking which comprise Process Education (PE) is the brainchild of Dr. Daniel Apple, President of Pacific Crest Software, Corvallis, OR. Much of what I have been able to accomplish in my own teaching results from my participation as a learner, mentor, and author in the many teaching institutes which he has sponsored.

"Process Education encompasses the philosophy that learning, thinking, problem solving, communicating, assessing, and teamwork are processes to be developed and continually improved by students as they construct knowledge. Process Education incorporates cooperative learning, discovery-based learning, journal writing, and extensive assessment." [1, 5]. I have successfully applied the processes mentioned in the last sentence to the teaching of Computer Science.

The driving force of process education is its focus on student learning. This results in a revolutionary way of viewing teaching. Instead of the traditional lecture approach, a teacher whose primary concern is student learning must first come to believe that every student can learn, and that a teacher's job is to facilitate learning by helping students develop learning skills to such an extent that they become capable of designing their own learning processes. We call this type of education "learning to learn." [2, 12]. If our colleges and universities can graduate students who will become life-long learners, we will be meeting the challenge that modern society has set out for us.

One of Dan Apple's models that I find helpful is the labeling of stages in the learning process as follows:

1. Trained: Ability to perform tasks without understanding.
2. Learned: Gaining understanding by being told. Rote learning.
3. Learner: Discovering concepts in a controlled setting.
4. Enhanced Learner: Asking own critical thinking questions.
5. Self Grower: Ability to develop own learning processes.

These stages fit the learning of Computer Science especially well. Often workers can be trained to use a software package without understanding what makes it work the way it does and often without even wondering about this question. Sometimes, beginning students in computer literacy courses fall into this role. Of course, when another release of the package arrives, or when they are forced to use a different package to accomplish their task, they are lost.

One way to help students understand the concepts underlying their work is to tell them. The lecture method of teaching has perfected this type of rote learning. We tell the students what they are to know. They write it down in their notebook. We ask them to repeat it on a sequence of quizzes and tests. The problem with this type of learning is that it is very short term. It has been verified that five years after graduating from college, only five percent of what a worker needs to know to do her job has been learned in college. The other 95 percent must be learned on the job. If a student has been a passive learner in college she will be ill-equipped to learn on the job.

A student who has had to discover the concepts she has learned in college will have developed skills which will help her to become a life-long learner. Discovery learning is often a closely guided process in which the facilitator (teacher) prepares a model and set of critical thinking questions which students work through in teams to gain mastery of the concept, tool or process being modeled. Closed lab activities are examples of discovery learning. If the lab is a means for students to practice what they were taught in lecture (Lecture Centered Learning), they find it hard to move to the "learner" stage. On the other hand, if lectures are used to further explore concepts discovered in the lab (Lab Centered Learning), students move much further along the path to becoming life-long learners.

Students who have learned to ask their own critical thinking questions are ready for independent research projects. Many computer science students do independent projects in groups and become enhanced learners in this way. They also gain experience working in teams and corporate recruiters value these teamwork skills.

The highest level of learning is called the self grower stage. Students who have reached this level realize that knowledge is divided into concepts (what is to be learned), tools (skills and resources needed to learn well), processes (step by step plans to accomplish a goal), and contexts (real life settings for the concepts, tools, and processes). Of these, the most difficult to master is the construction of processes, because this construction is a process in itself. Computer Science is well suited to developing this appreciation for abstraction since the development of algorithms is at the core of the discipline. With careful planning a good Computer Science program should graduate class after class of self growers.

In this paper, I will describe my own approach to teaching Computer Science including classroom management, how I have adapted four different courses to follow the process education approach, what successes I have acheived, and how students have responded.

Classroom Management of Process Education

In the introduction I have attempted to suggest the reasons why I have made the paradigm shift from a lecture mode of teaching to process-oriented mode. I found that students had difficulty moving from the Learned to the Learner stages of growth when I depended totally on lectures. First, I introduced closed labs into all of my courses and experienced the shift from lecture-centered to lab-centered teaching. Then, I began to assign team projects, especially in Systems Analysis and Design and Data Base, and I also began to use pairs or sometime triples in the lab.

At the same time, I attended some of the teaching institutes sponsored by Pacific Crest, and experienced the benefits of discovery learning. This is "a critical and unique aspect of process education in that it is the students and not the faculty who access appropriate information, and through suitably designed conceptual models or processes discover these new concepts by themselves." [1, 19]. The faculty member does not answer any questions during the discovery time. He or she asks critical thinking questions and encourages the students to "try it." In order for this approach to be successful, students should work in cooperative groups so they can build on each others' strengths and ideas and hence learn faster and better. "The characteristics of a cooperative-learning environment include individual accountability, high-level communication (face-to-face interaction), mutual interdependence, a focus on recognizing and developing process skills, ongoing reflection on the cooperative process, ownership of performance, and a shared community." [1, 19]

A final component which made this paradigm work for me is the emphasis on assessment. Each member of each student team has a defined role to play. I use four person teams and the roles (which different members assume each class) are: team captain, recorder, reflector and spokesperson. (See Appendix A for a description of the criteria for each role). The reflector makes periodic reports during the class. Each report contains an important strength exhibited by the team in the last 15-20 minutes which enhanced the learning process; an area most in need of improvement; and an insight about team learning.

At the end of each class, the team produces a report of the work products for the class period. These include the recorder's notes answering the critical thinking questions pondered by the team; the reflector's notes giving the strengths, areas for improvement, and insights; a grade on a scale of 5 which the team assigns to its own work during the class and a reason for the grade. I double their grade if it agrees with what I have observed about the team during class and with the quality of the team report. Otherwise, I give them the grade they assign themselves. It is seldom that I don't double the grade.

In addition, each student keeps a journal with an entry for each class which includes their individual assessment of how well they contributed during each class, their assessment of how well their team worked, an assessment of me as the facilitator, and an assessment of the discovery-learning activity. This forces each student to become aware of her learning process and keeps the focus of the course on learning.

A typical 75 minute class period is organized as follows: A brief review of what we learned last time and a few key points for the current class (5 minutes); a team quiz (10-15 minutes) on the reading assignment (if there is no lecture, students must read before class; those that don't come under pressure from their team members since they cannot contribute during the quiz); presentation of the quiz (10-15 minutes) by the spokesperson of one of the teams (after each answer the other teams are polled for alternate answers and I explain the correct answer); consulting period (5 minutes) where I answer any questions remaining from the reading (I get very few questions); a discovery learning activity (30-40 minutes) designed to let student teams answer critical thinking questions about a model which illustrates the main concept(s)/process(es)/tool(s)/context(s) of the class period; and end-of-class processing (5 minutes) where each team prepares its written class report.

If students have never engaged in this type of learning before, the first couple of weeks are very interesting. Many students, especially good students, are convinced that they will never be able to learn this way. I have to encourage them to try it, to experiment a little. I usually do not record any grades for that two-week period. Students receive two daily team grades which are the same for all members of the team: 10 possible points for the quiz, and 10 for the team report (see above for a description of the latter).

At the end of the first two weeks, I have an activity where the students choose the weight they wish to assign to each aspect of the course. Then the teams bargain among themselves until the whole class can accept a common set of point assignments. This gives them ownership of their learning and convinces them that I was serious when I gave them control of the learning process. I have never had a class assign points in a way that I did not agree with.

By the fourth week, the teams are cooperating fully, and the students are convinced they are learning better. All of the teams have gone through three stages: forming the group, resolving conflicts, and learning to problem solve together. Time pressure for quizes and activities sometimes facilitates this team development, but it also raises the frustration level. Sometimes, I switch groups halfway through the course, but if I have assigned semester-long group projects, I keep the same groups all semester. After the switch, team building is much faster. When I do so, I switch the group membership to give the students experience working with a number of different people.

I always try for the greatest diversity in each group. Good students work with poor students, men with women, extroverts with introverts, etc. Since the team teaches each other about the course concepts, everyone learns by explaining their ideas to the others.

I collect the journals at least four times during the semester and assign a grade on how conscientiously they were completed. I learn a lot about how the groups are working, how each student is learning and what I am doing well or poorly from the student journals. The students are so accustomed to assessment that they are quite free with suggestions about my areas of improvement as well as my strengths and insights about my interactions with them.

I also schedule a mid-course assessment where I ask the teams to brainstorm and come up with better ways to organize the classes. I almost always get really good suggestions which I can implement to improve learning. In this mid-course assessment, I tell them that each class is divided into three sections: preprocessing (consisting of the quiz and team presentation), processing (consisting of a consulting session and learning activity), and postprocessing (end-of-class assessment time). Note the similarity between this structure and the recommended structure for a closed laboratory session. I ask them to revise this class organization plan, including revising the time allocated to each task, in such a way that it improves the learning atmosphere in the class.

One of the suggestions they have made which I have incorporated is to let the other teams suggest answers to the quiz questions before I give my answers. We do this one question at a time. First the spokesperson of the presenting team gives her answer. Then I quickly poll the other teams. Finally I clear up misunderstandings and we move to the next question. This has resulted in significant improvement in learning efficiency. I get to see what the entire class understood from the reading and only need to explain what was misunderstood.

Another suggestion, which I have been unable to incorporate, is to schedule a similar time for review of the learning activity at the end of class. Often the teams work at different speeds through this activity and time runs out before I can get team reports.

An advantage of this way of teaching is that one is free to try new approaches with the knowledge that there will be almost instant feedback. Some of the best ideas come from the students. I become a role model of a self grower for them since they see me always seeking to develop better processes.

Specific Course Experience

The four computer science courses I have taught using process education are:

1. Systems Analysis and Design using Whitten, Bentley & Barlow [8];
2. C and Assembly Language, a computer organization course using Scragg [7];
3. Data Base using Kroenke [5];
4. Simulation using SLAM II by Pritsker [6].

All of these are project-oriented and to a certain extent lab-centered courses. They are listed in the order I taught them using the PE approach.

The most successful of the four was the Systems Analysis and Design course. The students were able to read and understand the entire 850 page text [8], and complete a semester long project during which they worked with real clients, developing specifications, performing an analysis of the current system with entity relationship and data flow diagrams, designing a new system, and implementing a prototype of some portions of the system. They worked in three person teams, meeting a sequence of 10 milestones and writing a fairly professional report by the end.

When I have taught this course in the past, I have found myself and the students bogged down in the sea of terms and tools. Making the students responsible for reading the text and experimenting with the software (we used Excelerator primarily) freed me up to explain what they did not understand and to prepare learning activities which gave them a real working knowledge of the concepts. My assessment is that learning was significantly improved and I was able to cover more material than in a traditional course. Sacrificing content for process is a frequent criticism of process education. The students and I were able to refute that claim.

The second course I taught using PE was one in which I had to teach languages (C and Assembly Language). How was I to encourage students to "discover" the syntax and semantics of two new languages? All the students had studied CS1 using Pascal (without objects) and some had taken CS2 using object-oriented Pascal. I was confident in their ability to design the solution to a problem and to write good user documentation, but I wanted to review these processes as well.

After consultation with Dan Apple, I decided to use the process of reengineering to teach these languages. I prepared a set of well written but poorly documented programs in C. The students worked in four-person teams and each team was assigned a different program to run, improve the documentation, and rewrite in Pascal. As each team finished its program, I rotated that team's better documented C program to another team until all three teams had seen and translated all the programs. By this time the students had a good understanding of C and were ready to do the first programming assignment also in teams. After that, each student was required to write her own programs.

I repeated this process when we got to assembly language, although this time the teams reengineered the assembly language programs into C. Otherwise, the class organization was similar to that described in section 2 above. I was not able to get as far in the course as I had hoped. This was the first semester I had tried to teach two new languages and I doubt if I would have succeeded as well using traditional methods. Next year, the students taking this computer organization course will have studied CS1 and CS2 under the object oriented C++ paradigm. I will have to introduce them to a functional paradigm when I have them use C. It will be interesting to see how readily these students make the reverse paradigm shift.

The course in Data Base was similar to the one in Systems Analysis and Design, except that the projects were more contained, textbook style. Many of the students took both courses. For some, it was the third or fourth course taught using process education. I began to get some rebellion against keeping a journal. One student reflected that she spent a half hour after each class "thinking of flowery phrases to relate what I have done. I do the same thing every time." I have experimented with different journal styles from free floating to answering a set of specific questions. I find that the most productive format for the students is for them to identify one strength, one area for improvement and one insight for their own individual work, the team's work, the facilitator's (myself) contribution, and that of the learning activity. I also have them write down 3 things they learned during the class and a process and skill that they have practiced. The purpose of the journal is to help students reflect on the learning process and not just on the concepts learned.

The final course, Simulation, is one I am currently teaching. It is the most mathematically oriented of the four and has the most challenging concepts. I always have trouble teaching the course, but using process education has proved as successful as any other approach. The students stay alert throughout each class and wrestle with the concepts under my guidance. Some of the activities I have planned have worked well and some have been too difficult. Students in the course learn to write FORTRAN programs as well as how to use SLAMSYSTEM, a simulation modeling environment.

I introduced a two hour closed lab in the Simulation course this semester and it has been very helpful in allowing pairs of students to design, code and run simulation problems under supervision. In past semesters students have presented some of the examples from the book during class, but working on their own problems was relegated to homework projects, requiring many hours in my office as they worked through difficulties. This semester, the practice from lab has helped the students become more independent while completing their projects.

Role of the Teacher

It has been quite an effort to make the switch to process education. Some teachers have done so gradually, but I made the transition cold-turkey. The first class I tried it in was Finite Mathematics and it was a disaster, primarily because I did not understand well enough the interactions of all the roles I needed to play. After attending two more teaching institutes, I have had significant success for the past two years, as indicated in the preceding section. My teaching evaluations have reached an all-time high, and students returning from interviews relate that most of the interview was devoted to their experiences in my courses. I believe that students, particularly the weaker ones, are learning more and gaining greater confidence in their ability to learn.

All this success is not without its price, however. I have spent long hours preparing quality learning activities, grading quizes, activities, journals, as well as the usual tests, projects, homework assignments, etc. Fortunately, as I get more practice with the methodology, I have to spend less time in preparation. I am hopeful that, with WWW access improving, repositories of successful learning activities for a variety of courses will become available. Perhaps, as textbooks are published online, the process education adaptations for courses using these books will be integrated into the text.

Let me describe the role of the teacher in the classroom during a process education-oriented class. For one accustomed to performing as a lecturer while students took notes and asked questions, it was quite a change for me to take on the roles of leader, assessor, facilitator, and evaluator real time in the classroom. This division of roles and the following description is due to David Hanson from the Chemistry Department at SUNY Stony Brook, and Daniel Apple from Pacific Crest Software [4, 10].

"As *leader*, the instructor develops and explains the lesson, defines the objectives (both academic and process skills objectives ...), criteria for success, expected behaviors, and establishes the organization [classroom management structure...].

"As *monitor/assessor*, the instructor circulates through the class to monitor and assess individual team performance and to acquire information on student understanding, misconceptions, and difficulties in collaboration.

"As *facilitator*, the instructor intervenes and asks timely critical thinking questions to help groups understand how they are functioning, why they may be having difficulty, and what they need to do to improve and make progress. By combining monitoring, assessing and facilitating, the instructor assures that all understand the assignment, that each group member is fulfilling their assigned role, that positive verbal exchanges are occurring, and that progress is being made. Such intervention provides [realtime] feedback, motivation, and reinforcement; teaches academic and collaborative skills; and guides students in the use of a problem-solving methodology.

"As *evaluator*, the instructor provides closure to the lesson by asking group members to report answers, summarize the major points, and explain the strategies, actions, and results of the group. the evaluator also provides evaluations to individuals and groups regarding performance, achievement, and effectiveness and shares general points with the class." [4, 10]

Conclusion

While the role of a PE teacher seems somewhat overwhelming, one can grow into it. An important aspect of process education for the student is that learning is fun. Facilitating process education should also be fun. After the initial shock, students appreciate the opportunity to learn this way. Constant feedback helps both student and teacher perform better. Process education requires that a teacher take risks, but the results are significant. All the students improve their learning process, not just the better students.

Not only do students develop learning skills, they have an advantage when it comes to finding a job. Employers seek individuals who excel as: 1) Quick Learners, 2) Critical Thinkers, 3) Problem Solvers, 4) Communicators, 5) Professionals Knowledgeable in Their Field, 6) Team Players, 7) Self Starters, and 8) Creative Thinkers. [3, vii] These skills are the ones practiced while experiencing process education, and job recruiters compete for students with these skills.

In summary, process education seems a natural for teaching computer science because CS students have the potential to move through all the stages of learning rather quickly. The closed lab component of many computer science courses already utilizes some of the PE processes. Computer science faculty are used to taking risks since the discipline is in such a state of flux. In my experience the change to process education is a risk well worth taking.

REFERENCES

1. D. K. Apple and W. Duncan-Hewitt, A Primer for Process Education, Pacific Crest Software, Corvallis, OR, 1995.

2. D. K. Apple et al., Foundations of Learning, Pacific Crest Software, Corvallis, OR, 1995.

3. D. K. Apple, S. W. Beyerlein, M. A. Schlesinger, Learning Through Problem Solving, Pacific Crest Software, Corvallis, OR, 1992.

4. D. M. Hansen and D. K. Apple, Process - The Missing Element, Pacific Crest Software, Corvallis, OR, 1995.

5. D. M. Kroenke, Database Processing, Macmillan, New York, 1992.

6. A. A. B. Pritsker, Introduction to Simulation and SLAM II, Systems Publishing Corporation, West Lafayette, IN, 1995.

7. G. W. Scragg, Computer Organization: A Top-Down Approach, McGraw Hill, 1992.

8. J. L. Whitten, L. D. Bentley, V. M. Barlow, Systems Analysis & Design Methods, Irwin, Burr Ridge, IL, 1994.

Appendix A

Each class will have group exercises. Teams should rotate the following roles so that each member gains experience in each role.

Team Captain

- Keeps the group on task and makes sure everyone is having fun;
- Ensures that the group accomplishes the task within the allotted time;
- Encourages full participation by each group member;
- Ensures that all team members can articulate what has been learned;
- Ensures that the other group members perform within their roles;
- Represents the group in all interactions with the instructor.

Recorder

- Records group roles and instructions at beginning of activity;
- Takes notes of important points which come up during the activity;
- Ensures her words accurately reflect group consensus. Checks wording with the group when necessary;
- Gives the group report orally at the end of the activity unless the instructor or team leader indicates otherwise;
- Prepares a written report of the group's decisions and discoveries during each activity.

Reflector

- Watches group process: what is going well, what is needs improvement, what can be done to improve the process (use constructive criticism);
- Reports her observations to the group when needed (at least once every 15 minutes) to help make the work go forward and help the group to function better. The report should contain: (1) the group's most important strength since the last report; (2) what skill the group most needs to improve on; and (3) an insight into how group learning can be improved. These reports should be written in the reflector's journal and time stamped;
- Prepares a written assessment of group process during the class or lab and checks it out with the group during the end-of-class assessment period. This will include:
    - How well the group performed as a team and how well each member performed her individual role;
    - two greatest strengths, two areas most in need of improvement, and the major insight gained from group's activity during the class;
- Reminds the team leader of her duties if necessary.

Spokesperson

- Represents viewpoints held by a majority of the team members.
- Retrieves information from various sources (computer, text, manuals...).
- Encourages risk-taking and critical thinking.
- Delivers required oral reports about the activity.

Note that all group members must fully participate in the learning exercise while performing their roles. All must really want the team to succeed.

PERFORMANCE CRITERIA FOR GROUP ROLES

Team Captain

1. Keep the process enjoyable and rewarding;
2. Keep all team members performing within roles;
3. Keep the team focussed;
4. Keep all team members involved in the problem-solving process;
5. Ensure that all team members can articulate what has been learned;
6. Control the process: keep it systematic;
7. Strengthen the weakest link;
8. Time management;
9. Stress management;
10. Active learner and contributor;
11. Overall team performance;
12. Plan time for various tasks;
13. Set up meeting times and places, meeting length, deadlines;
14. Act as the objective internal mediator when interpersonal conflicts arise;

Recorder

1. Record names and group roles at beginning of class;
2. Record instructions at beginning of task;
3. Quality of listening and recording skills;
4. Legibly document the process, group decisions, and discoveries in the group Learning Journal;
5. Active learner and contributor;
6. Ability to control information flow;
7. Ability to rearticulate concepts in alternative forms;
8. Ability to integrate and synthesize multiple ideas;
9. Ability to diagram and draw pictures;
10. Create and communicate algorithms;
11. Prepare the report for handing in at end of class;

Reflector

1. Ability to rephrase or reframe observations into constructive criticism;
2. Observation skills;
3. Report strength, area for improvement, and insight in the Reflector's Journal and verbally to the group at least once every 15 minutes;
4. Provide information about group interactions;
5. Provide information about the group process;
6. Active learner and contributor;
7. Intervene with observations about the process and strategies for change;
8. Remind team leader of duties;
9. Acquire sufficient evidence of behaviors and documentation to permit fair judgements to be made in the event of mediation and conflict resolution.

Spokesperson

1. Listening and communication skills;
2. Observation skills;
3. Experimental skills;
4. Retrieving information from various sources;
5. Active learner and contributor;
6. Planning and management skills;
7. Critical thinking;
8. Collaborating;
9. Synthesizing;
10. Risk taking;
11. Computer skills;
12. Giving oral reports;