22

Teaching Systems Development*

Joan Greenbaum
Lars Mathiassen

Abstract. There has been a great deal written about curriculum for teaching systems development, but very little said about the methods and practices of the teaching process itself. This article, in the form of an open letter to colleagues, addresses itself to the problems and experiences we have had in teaching systems development. Specifically, it focuses on the contradiction between education and experience and suggests some metaphors and methods for better integrating experience into the learning process.

> *"Sometime look at a novice workman or a bad workman and compare his expression with that of a craftsman whose work you know is excellent and you'll see the difference. The craftsman isn't ever following a single line of instruction. He's making decisions as he goes along." (Pirsig 1974, p.149).*

Dear Colleagues,

Not long ago we received a letter from one of our colleagues. At first we set out to write back immediately. But in doing so, we realized that we first had to discuss the issues involved, and even worse, we had to reflect back on our own experiences as teachers. Our colleague wrote as follows:

> *"As you well know I have been teaching systems development for years, but the more I do it, the more confused I become. Only recently, I concluded that it's a little like trying*

* Published as: Zen and the Art of Teaching Systems Development. J. Greenbaum & L. Mathiassen. In: Computers and Society, ACM, Vol. 20, No. 1, 1990.

> *to teach sex education to virgins! Though, thankfully, I've never tried to teach sex, I imagine that teaching virgins too much* about *sex might do more harm than good in relation to their future sexual practice.*
>
> *In our profession, we seem to saddle our students with so much talk of theory and especially methods, that they become confused when they actually have to apply them. While we compensate for the students' lack of systems experience, with the old "stand-by", the case study approach, we often fall short of being able to actually integrate theory, method and experience.*
>
> *I feel that, as teachers, we ought to be able to make more out of the basic tension between* education and experience. *You have both been teaching systems development for a long time. Do you have some constructive examples and experiences that could shed some light on the basic problems of teaching and motivating in our field? Or more specifically, can you suggest ways to better integrate student experiences into the* way *we teach systems development?"*

The letter certainly provoked us. But after several discussions we still were not able to come up with a satisfactory answer. As we continued our talks we found ourselves deeply engaged in reflecting on our own teaching experiences. A process of reflection that has led both to changes in the way we teach and, we think, a heightened awareness of the tension between education and experience. Indeed, the letter propelled us to begin this writing process.

There has been a great deal written about *what* is to be taught—curriculum, case books, and the like. Here, instead, we want to talk about *how* we teach systems development, keeping in mind, of course, that this issue has to be seen in relation to what we teach. We hope that our thoughts act as a catalyst to others to continue the discussion. For, as surely as there is a need to constantly revise the curriculum there is a strong need to consciously reconsider the way we teach.

Seeing and acting

For years we have used case studies and field work to integrate practice in the educational setting. We still believe that this is a

fruitful and necessary approach. It is, however, insufficient sometimes as a way of exploiting the contradiction between education and experience.

Usually, early in their case work, the students wander in to ask us questions like: “I'm not sure what the problem is ?”, “Am I using the right method?”, or most commonly, “Is this right?” Coming from a background in Computer Science, as most of our students do, they are trained to look for *solutions*. In their previous experience they are used to being confronted with specific problems and types of solutions. This is even more pronounced with students who are trained in mathematics as well.

The students want certainty and step-by-step clarity in their actions. And why not? It is so comfortable compared to the chaos and uncertainty of most systems development situations. We think, that perhaps, we have been using case studies and field work in too narrow a fashion. Instead of simply using case studies to *apply* the classroom concepts, we want to emphasize the *type and quality of all experiences;* beginning with classroom learning as a form of experience.

It is in no way wrong, of course, to solve problems and to look for solutions. But the point is that systems development as a *process*, requires *problem-setting* as well as problem solving (Lanzara 1983; Schön 1983). A focus on problem-setting shifts the emphasis toward viewing the context and environment of the issue; not just the problem at hand.

As teachers we hope that students learn to look openly and critically at situations and to act in uncertain and even chaotic situations (Lanzara 1983). Likewise we believe that they should not take problems for granted, but should learn to *interpret situations* (Checkland 1981). And, of course we urge students not just to follow methods, but, instead, to learn to find their way through landscapes that are only partly known to them. In short, they should learn to see and act within the experiences open to them.

All of this points toward helping students handle uncertain and complex situations. Yet, in practice, we send these poor students out to do case studies and field work, laden down with an almost endless supply of tools, methods, and interesting theories. To some extent this is part of the idea with field work, but in most cases they've got so many things to think about that they can hardly

see nor act. (We wonder if this is also true with virgins who have studied about sex!)

To make a lot of long stories short, we began to re-frame the teaching process to better reflect the chaos and uncertainty of the systems development process. Now, don't jump to the wrong conclusions, for you certainly know us well enough to know that we are not fanatically pursuing a path to chaos, nor are we anxious to join the zeolots of Zen or the mechanics of motorcycle maintenance. We do, however, feel that the chaos and instability inherent in both teaching and systems development are a ripe source of experiences with which to shape the teaching of systems development.

The process of teaching is, after all, a lot like the process of systems development. We never really know what the end result is going to be like and how it is going to be used! We can certainly not expect students to become competent systems developers through a series of step-by-step instructions, any more than we can do reasonable systems development in this way. Teaching, as we know intuitively, is helping students make their own decisions. And it is this process—the process of exploring and testing—that can give students a focus on *experience* and the *context* of experience that they are missing.

Student experiences

Last semester, in an advanced seminar on Systems Development, we noticed the not uncommon problem of “lack of student discussion”. It wasn’t that the students were lacking in ideas, but rather, there seemed to be a silent game of “I’ll speak after someone else does” going on. One of the students raised the problem in class and we tried a brief discussion to see if we could find out what was causing it. But like many such teacher-student dialogues we didn't get very far: we asked them what they thought the problem was; they responded that they didn't know; and there the issue rested. We are sure you're familiar with this problem, and are probably laughing at our naivity to think that we could simply put the problem of “lack of discussion” to the students and then have them solve the very issue that was causing them a problem!

It wasn't until after the course was over that one of the students came in to talk about it. She was responding to Joan's suggestion that students reflect on the highlights and bad points of the

course and come in to discuss it. The student set the issue rather clearly: “Didn't you”, she said, “tell us that communication is one of the key issues in systems development?”. Certainly, we had. “Well then”, she continued, “this problem of not talking in class was certainly a big communication issue”. We admitted that it was. “So”, she concluded, “why didn't you stop the normal process of lecture, exercise, etc., and let us deal with this problem. After all, since we had raised the problem, and knew it to be an important part of the methods you were teaching, we should have come to grips with it right then and there”.

On reflection, we were probably a little afraid of opening the “Pandora's box” on this issue. The student was right: We should have let the students tackle it head-on rather than returning to the “points” that we had planned to teach them. Here we were presenting theories on communication and its relevance to the systems development process and they were afraid to *talk* about it! And we were afraid to stop the *planned process* of the course in order to get at the problem. Had we taken up the problem that the students had set for themselves (and us), we could have re-framed the problem to let their own experience guide them.

Instead, we remained stuck in the years of practice we have had in teaching systems development and preparing outlines and notes. It's certainly easier, as a teacher, to talk about what you know best, than to wander into the unknown of student problems! In discussing the need for an “expansion of reason” and new forms of rationality, Pirsig, reminds us that our fear of this unknown “is comparable to the fear people once had of falling off the edge of the world” (Pirsig 1974, p. 151). Yes, indeed, teaching in an environment where students set their own problems is a little like “falling off the edge of the world”.

The question of certainty and control over the unknown is deeply rooted in the methods and practices of both teaching and systems development. Both follow the perspective of “technical rationality” which has pushed Western science to flow rapidly to the goal of finding technical solutions to all problems. While it may be comforting to exercise control over the teaching process, control and certainty are probably no more *real* in the classroom than they are in the offices of a systems developer.

Without going into a long history on this point, it is worth noticing that this path of “technical rationality” has led us to make

false dichotomies. Dichotomies that force us to build walls between ideas and race towards solutions without stopping to reflect and experience on the nature or setting of the problem (Greenbaum 1987). One such dichotomy is the gender bias of science and technology. As Evelyn Fox Keller (1985) points out in *Reflections on Gender in Science,* the history of science and technology has been built on a base that sees science as objective and male, while nature is seen as subjective and female. Fox Keller explains that these myths, that is the beliefs about science as male, and nature as female, strongly influence the way we practice science. For the history of science and its offspring, technology, can be told as the history of control or mastery over the unpredictable, chaotic conditions of nature. Even if we accept these dichotomies as false, our practices are rooted in patterns that value control and objectivity over uncertainty and subjectivity.

A great deal can be gained by becoming consciously aware of these false divisions, and using the experiences of systems development to help actors guide and control their own route through these chaotic and uncertain waters. The same holds true for teaching. If students are encouraged to set their own problems, and be aware of their experiences as they do so, then they are hopefully, taking steps toward managing both the learning process and the systems development process, as well. But so much for theory, the question our colleague asked was how can we do this!

Relevant experiences

Within the teaching process, there are at least two ways that student experiences can be made relevant to their future work as systems developers.

First, as we have hinted at already, every systems development effort is a learning process for the actors involved. The actors, whether students or experienced analysts, must learn about the traditions and customs of the user organization, as well as learning about the technical and organizational options. Imagine for a moment, a systems analyst who cuts him or herself off from the user environment—as absurd as an anthropologist with no notion of culture!

Systems development is a process of learning, imagining, experiencing and creating. This ongoing learning process takes place

within worlds of different traditions and practices. The systems developer ventures again and again into new and unchartered user worlds, each time to emerge with some concrete vision. He or she is, perhaps, an anthropologist of the future, supporting people in their imaginative design and creation of future use situations. Experiences build upon experiences, but the key is the developers ability to learn from the process itself. In the teaching of systems development, we need to use the teaching process as an arena for letting students experiment with how to design and participate in *ongoing learning activities.*

The second suggestion flows from the first. Systems development efforts typically require intensive management and coordination activities to cope with the uncertainties and complexities involved. Most systems development efforts cannot be managed effectively by a project manager alone—they need the cooperation and active involvement of all of the actors. Systems developers need to know how to plan, evaluate, reflect upon and intervene into the work processes they participate in. The experienced systems analyst, or "craftsman", in Pirsig's terms, is deeply involved in self-management and keenly aware of planning and cooperating with others. The implications of this for teaching systems development are clear: we need to set projects and exercises as places for *really* letting students experiment with project organization, self management and cooperate planning.

Of course, you may say, we already try to do this with case studies, projects and field work. Students, like systems developers (and the rest of us, for that matter), are, however, primarily evaluated on the final product. This reality, unfortunately, stands the test of time. As teachers, we can certainly do more to help students guide themselves through both the learning process and the development of social and organizational planning. Think of all the times that students have come in to complain about "how badly their project was going because they couldn't work together", or how the deadline for the project forced them to "just write something up to get it done on time". We suggest that these are moments to engage in substantial dialogues.

Metaphors and methods

Recently, we organized a course around short exercises that were

designed to involve students in experimenting with the ideas we had presented. In one such exercise, the students were to evaluate the difference between *description and analysis* by looking at a group of pictures and describing what they saw. In concept the exercise was a good idea, but in practice, we made a critical mistake: we first *told* the students what to look for and then asked them to do it.

When we asked the students to reflect on this method of teaching they quickly pointed out the problem "Why", asked one, "did you think that you had to tell us what to look for?" Another put the problem more bluntly saying, "Did you think that what you had to say was more important than the way we found out to do it ourselves?" It wasn't the exercises that were at fault, but the way we used them.

Maybe one of our greatest mistakes is that we tend to think of ourselves primarily as teachers giving courses to students. In the teacher-student metaphor our task is to teach students about systems development, and our primary means are methods and theories as described in books; and practice, as represented in case studies and the like. But there are other metaphors to use when thinking about the learning process; metaphors that guide us to reflect on our teaching practices. Here, we will look at two examples, but we invite you to further the discussion with images from your own experience.

Master-Apprentice

The first metaphor we have borrowed from Pirsig. It is the relationship between master and apprentice, and through this we get a different picture. In the master-apprentice case, it is our task to organize experiments and exercises to demonstrate *by doing,* and to constructively guide the performance of the apprentices.

C. W. Mills once defined methods as ways of asking and answering questions, with some assurance that the answers are more or less durable. He argued that method can only be imparted to beginners "by conversations in which experienced thinkers exchange information about their actual, informal ways of working" (Mills 1980, see also Naur 1985). In the master-apprentice metaphor we believe that what we do, and how we perform is the primary concern. We emphasize our own actual, informal ways of working, at least those parts that are relevant to systems development, and

from these we engage in conversations leading to methods and practice.

To return to our colleague's letter, we may have been misled by the suggestion that teaching systems development is like teaching sex to virgins. *By looking at our students as virgins we imply that they have no real and relevant experiences.* On the contrary, they certainly have experiences; what they need are ways of working to help them reflect and apply these experiences. Lars often tells the story about how his experiences as a boy scout helped him to become a teacher of systems development! While the experience sounds a little far-fetched, he always goes on to explain, that learning to work with others and investigate things for himself was a cornerstone of his boy scout years.

Well, scout leader or not, we find ourselves in double roles as teachers and as process-consultants. By looking at student experiences as a well-spring of the teaching process we can ease ourselves into the master-apprentice role. Admittedly, this requires that we, as teachers, are willing to *expose our own working habits* and have the guts and competence to let students become, as Mills suggests, "self-conscious thinkers", instead of obedient followers of pre-selected methods and theories. Yes, yes, we all say—of course, we are willing to try. But doing it, well, maybe that's another matter.

Material Matters

The second metaphor is borrowed from a colleague who commented on a working draft of this paper:

> *"I know that you will agree with me, when I say that teaching and learning systems development demands a certain amount of* enthusiasm *about the problems one has to deal with. I don't think that the need for enthusiasm is special for systems development (or for sex, for that matter), but the need is specially important exactly because of the chaotic and uncertain nature of the process of systems development."*

So a discussion about enthusiasm led us to look at a second metaphor—the relationship between the student and the material he or she will learn. In this student-material metaphor our task is to help the student get involved or excited by the material. The selection of material—such as cases, examples, theories and issues—is central for students to generate enthusiasm and thus take on the responsi-

bility for their own learning process. After all, any master-apprentice relationship would break down if the apprentice didn't feel connected or involved with what they were learning.

Here the signposts for developing teaching practices are a little clearer. If we borrow some of the time-tested ideas of Pablo Freire (1968), we know we can rely on student experiences to *motivate* the learning process. For example, looking at systems development as a creative activity leads us to help students try examples from other design disciplines. In one situation, a group of our students who were 'stuck' for an idea to start their project, remembered that the Danish designer and architect, Poul Henningsen, had much to say about the design process. Reading Poul Henningsen's ideas about design got them going on trying to formulate their own project. Another student group became fascinated with the concept of Postmodernism and used this as a starting point to push their design ideas.

Pirsig also pushes us in this direction:

> "*What's wrong with technology is that its not connected in any real way with matters of the spirit and of the heart*".

Encouraging enthusiasm in the student-material relationship, is easier said than done, of course. As teachers we can't create student enthusiasm, but hopefully, we can use our own to help them find interest in the material they are studying.

Three of our students recently began work on their Masters thesis in systems development. After several months of careful planning they finally came in with a well-defined topic (a process everyone knows to be very difficult in any discipline). They were obviously quite pleased with their progress, but one of the first questions they asked was "Do you think that this is a *good* topic?" "Yes", we responded, "but how do you *feel* about it—are you excited about doing it?" Luckily they were, and the thesis is now well underway. But, if as teachers, we were to concentrate more on the topic, than on the students' feeling for their topic, we would become as unstuck in time and space as the notion of art and technology as belonging in two separate worlds.

We began this paper with a quote from Pirsig on craftsmanship, but we left out a rather important part. He continues: "Sounds like art", the instructor says (about craftsmanship)."Well, it is art, I say. This divorce of art from technology is completely unnatural. Its

just gone on so long you have to be an archeologist to find out where the two separated" (Pirsig 1974, p. 148).

Teaching, like systems development, has many unknown results. Systems developers may think that they have carefully planned the outcome of a system, but its use, in practice, will be different. So it is with teaching systems development. We may root our plans in solid curriculum, but the outcome is only successful in the way students apply the concepts and approaches within the realm of their experiences.

We think that it is useful to look at teaching *and* systems development as both art and technology. In fact, we think it is possible to begin the process of integrating these false dichotomies by weaving teaching and systems development together. The emerging pattern can begin to help us (and our students, of course) use the experiences in the learning process to learn more about the systems development world. For the tapestry of the systems development world is neither pure art nor technology, but some imaginative creation of the two.

References

Checkland, P. (1981): *Systems Thinking, Systems Practice.* Chichester: John Wiley.

Freire, P. (1968): *Pedagogy of the Oppressed.* New York: Herder and Herder.

Greenbaum, J. (1987): *The Head and the Heart.* Computer Science Department, Aarhus University.

Keller, E. F. (1985): *Reflections on Gender and Science.* London: Yale University Press.

Lanzara, G. F. (1983): The Design Process. Frames, Metaphors and Games. In U. Briefs *et al.* (Eds.): *Systems Design For, With and By the Users.* Amsterdam: North-Holland.

Mills, C. W. (1980): On Intellectual Craftsmanship (1952). *Social Science and Modern Society.* Jan.–Feb.

Naur, P. (1985): Intuition in Software Development. in H. Ehrig *et al.* (Eds.): *Lecture Notes in Computer Science 186.* Berlin: Springer-Verlag.

Patton, M. Q. (1980): *Qualitative Evaluation Methods.* London: Sage Press.

Pirsig, R. (1974): *Zen and the Art of Motorcycle Maintenance, An Inquiry Into Values.* New York: Bantom Books.

Schön, D. (1983): *The Reflective Practitioner. How Professionals Think in Action.* New York: Basic Books.