Int. J. Engng Ed. Vol. 14, No. 1, p. 14–23, 1998
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Problem-based Learning: A Vital Step Towards a New Work Environment*

ROGER G. HADGRAFT
Department of Civil Engineering, Monash University, Clayton, Victoria 3168, Australia

As educators, we dream of highly motivated students who devour our courses with relish, and who are then able to competently apply what they have learned. Similarly, employers wish for highly motivated employees who will give 100% effort to their work. These are very similar requirements. In Australia, the Federal Government has offered $80M for each of the last three years, to encourage universities to develop quality assurance procedures. Although some good has come of this (e.g. an Education Policy at the author's university), little has really changed in the classroom. We still have not tackled motivation, probably the number one factor in student performance. This paper considers some of the literature on quality management, particularly of quality management, management of software teams and building creative organisations. It shows that problem-based learning (PBL) is a first step towards creating a new culture in university departments based on trust and respect for student contributions. It suggests that our departments will be the better for such a change, but that such a change will not be trivial.

INTRODUCTION

THERE IS a tendency in educational literature to neglect the role of students as people. Even when students are an important part of the process (e.g. in problem-based learning, PBL), most of us still keep thinking of them as external to the department, with little to contribute to the department at large. We may well have revolutionised the learning environment through active student learning, but has this spilled into the department's activities as a whole? Are we missing an opportunity to involve these hundreds of young minds in our research and community activities?

This paper draws on management literature in quality, creative organisations and computer software teams to suggest how we could improve our university departments. The focus is on new attitudes to working with our students. We need to see ourselves as managers of learning and research, and when we do that, new possibilities for collaborative work arise. Such changes will ultimately lead to more productive and happier academic staff—a win-win situation.

The paper shows the similarities between what many are doing in Problem-based learning and what people-oriented management theories are trying to accomplish in the workplace. It shows that with some small additional steps, we can satisfy both the learning objectives and the management objectives, moving our departments into a new mode where undergraduates are seen to be as much a part of the department as the academics. With this significant increase in personpower, departmental productivity must increase, and all involved should be happier—an important factor in greater productivity.

QUALITY SYSTEMS

Is there anything to be learned from the quality management systems movement and other management theories? After all, quality programs have been in place in manufacturing for 40 years, and in other organisations for at least 20 years, often with spectacular success, e.g. in Japan. We do not wish to view universities as some form of production line—that would be inappropriate to both students and staff. However, some broad principles might give us a new perspective on our educational activities and will assure us that we are moving in the right direction.

An important part of any quality program in industry is **worker empowerment,** where workers are given responsibility for improving the productivity of their working environment [1]. This is often achieved through informal or formal teams who identify problems, prioritise them, seek solutions and see that they are implemented (sound familiar?). An essential step is middle management passing this responsibility to the workers. (This is not an easy process for many of these managers). PBL is similarly team-oriented with students empowered to identify their learning needs. Many academics find yielding this power similarly difficult.

Problem-based learning

PBL is well described in books such as Boud and Feletti [2] and Ryan [3], and it has found acceptance in many disciplines, in many different parts of

* Accepted 15 August 1996.

the world. It is an approach with many variations, but the key ingredients are [4, Table 2]:

- **active learning** through posing questions and seeking answers;
- **integrated learning** by tackling problems for which knowledge of several sub-disciplines is necessary;
- **cumulative learning,** by a succession of increasingly more complex problems, working up to those which would be typically handled by a young professional;
- **learning for understanding,** rather than for the retention of facts, by providing time for reflection, by frequent feedback, and by opportunities to practise the skills which have been learned.

PBL has other benefits, which many of us see as essential to our graduates in their working lives, skills in [4, Table 1]:

- adapting to, and participating in change;
- problem solving in unfamiliar situations;
- reasoning critically and creatively;
- using a systems, or holistic approach;
- collaborating productively in teams;
- identifying one's own strengths and weaknesses;
- committing oneself to lifelong learning as a means of addressing the problems.

Quality and problem-based learning

Interestingly, many of these things appear also in the quality systems literature, for example, Deming [5] who proposed the 14 points for implementing quality procedures. The author and one of his students has discussed the correspondence between Deming's 14 points of total quality management (TQM) and PBL in some detail [6], and these issues are summarised in Table 1.

Some key points are:

- There must be **committed management**—this is the hard part!
- A negotiated **departmental vision** of what the department is about is required.
- **Training** in learning principles for both staff and students is required.
- **Trust and respect** must be built between staff and students.
- Adopt **problem-based learning** (or similar).
- **Eliminate** unrealistic goals or formal exams.
- Have students help build **learning resources** for others to use.
- Have students actively involved in **research projects**.
- Build a **team approach** in both teaching and research.
- Involve staff from **other departments** and faculties.

'Customers'

Much has been written about students as *customers of education*, but little about their role as workers. We all know that students ultimately do the hard work—learning. Teaching is easy by comparison! If we are to apply quality management principles, then, as middle managers, we academics must pass responsibility for improvement to the workers—to the students. Of course,

Table 1. TQM and PBL compared

	TQM	PBL
1	Constancy of purpose.	A shared (negotiated) vision for the department is required.
2	Adopt quality as a guiding principle.	Management must be committed; changing academics is the hard part!
3	Cease dependence on mass inspection.	Reduce the number of formal exams; ensure quality rather than test for it; get it right the first time. Problem-based learning encourages the best work, rather than minimal effort.
4	Do not award business on the basis of price tag.	Choose a teaching method that maximises the benefits minus the costs—problem-based learning may not reduce staff time, but it could increase total output (e.g. publications).
5	Improve constancy.	This is taken for granted in the research sector but with active student involvement it could also happen at the under graduate level; students could contribute to the development of resources for subjects.
6	Institute training and education for all members of the organisation.	It is essential that staff and students are trained to get the most out of learning by understanding learning principles.
7	Institute leadership.	Heads of department must be leaders in teaching as well as research; this is probably the greatest weakness we have with appointments being largely based on research reputations.
8	Drive out fear; build trust and respect.	Trust and mutual respect will generate outstanding work from our students. It is an essential part of problem-based learning.
9	Break down barriers between departments (e.g. teaching and research).	Make undergraduate courses closer to the current research activities through problem-based learning case studies, data analysis, etc. and actively involve staff and students from other departments.
10	Eliminate slogans.	Slogans are replaced by trust and respect and recognition that much of what is wrong with student performance is a management problem.
11	Eliminate numerical quotas.	Work to eliminate restrictive numbers such as credit point restrictions and restrictive assignment requirements and deadlines.
12	Ensure pride of workmanship.	Team ownership is a natural part of problem-based learning.
13	Vigorous education and self-improvement.	Problem-based learning prepares students for lifelong learning.
14	Take action.	The problems lie with management; vision is required.

this is exactly what PBL and other forms of active student learning have been doing for many years. We all have anecdotes, of the kind one reads in the quality systems literature, of students whose attitudes have been turned around by a PBL course—from unmotivated, to highly motivated. As in industry, we need respect on both sides to make this synergy work.

Staff need to feel that they are also winning from this process. Student involvement in staff research programs will benefit staff, students and departmental productivity. For example, undergraduates have computing skills sorely needed by staff. With these sorts of incentives, we have a win-win situation. Students get a more rewarding learning environment, and staff increase the department's research output.

Further, the way in which Quality programs have been applied in industry suggest that it is important to know your customer. In higher education, we must meet the needs of various 'customers' (students, employers, staff, the professions, the community), and all of these should have some influence on the content and direction of our courses. Thinking about the students alone, that means that students must be involved in setting course content and process. Too often, student involvement in departmental committees is reduced to tokenism. The real decisions are made in working parties and merely ratified at the departmental level. Students seem to be rarely invited onto these working parties. This is another outcome from our quality systems thinking.

The ideals of the quality systems movement are clearly aligned with the aims of problem-based learning. The key is *worker empowerment* and corresponding changes in *staff* (*management*) *attitudes*. For those already involved in problem-based learning, this should not be a problem. Extending empowerment beyond the learning tasks to embrace the department's research goals is the key to bringing the students on board as co-workers. This should increase departmental productivity.

SUCCESSFUL QUALITY TEAMS

DeMarco and Lister [7] wrote a people-oriented book for the software industry—an industry based on intellectual activity like universities. It recognises that the reason that software projects fail is more often linked with sociology than with technology (p. 4). By extension, it is probable that students do poorly in education for sociological reasons.

The book is interesting because of its focus on people, and particularly its focus on teams, which both problem-based learning and the quality movement also see as important. It is always interesting to see what insights one can gain from another discipline which is doing similar things.

Chapter 4 is about quality systems, and refers to some of the quality management literature (e.g. Crosby [8]). What is interesting here is the assertion that people tie their self-esteem to the quality not the quantity of their work. Once self-esteem is threatened, emotions can run high. DeMarco and Lister go on to indicate how organisations often do not value quality, and hence they threaten the self-esteem of their workers. Of course, Deming has said similar things (pride of workmanship) [5].

It is curious then that organisations, including universities, work so hard (perhaps unconsciously) to make quality so difficult. In an educational context, deadlines are the most obvious. We cannot guarantee that students will be ready to submit an assignment on a due date. Yes, deadlines are important in industry, but the author's experience is that if we are prepared to be flexible, the quality of the assignments will be higher. Students should be involved in determining any deadlines (see later).

Some educators say that this is unfair to students who do meet the deadline. There is an element of truth in this. Given another week, they may have been prepared to polish it further. This can probably be handled by a reward/penalty system (for early and late submission). A student can then determine the optimum time to submit for them. The author has had students submit weeks late, grateful that the work would be assessed—not realising that that was the point of the assignment! A later section on productivity makes some more comments on this.

Management and teaching

DeMarco and Lister refer to the need for good managers in several parts of the book. They indicate that a good manager needs to be someone who creates the right environment (physical, sociological and financial) for quality work to happen. DeMarco describes an incident when he was a young programmer (p. 34), when his manager fetched him a cup of soup on a day when he had to make an important presentation and he was quite ill. He asked how she had the time to do such things (as a manager), and she smiled and said: 'Tom, this is management'.

Similarly, great teachers create the environment for learning (e.g. the film *Dead Poets Society*). They know that the real work must be done by the students, and creating a supportive environment is the way to do that [7, p. 93]:

- get the right people;
- make them happy so they don't want to leave;
- turn them loose.

This is in direct opposition to many managers (and teachers) who believe in the big stick approach. The productivity figures in the next section suggest that this is of little value.

Productivity and motivation

Programming is an intellectual activity, like learning, and DeMarco and Lister are at great pains to indicate how to build physical and sociological environments which support it. Chapter 5 discusses productivity and attempts to dispel the myth that people are basically lazy and will try to do as little as possible. There is often a similar feeling among staff that students want to do as little as possible. Prosser lists this as an attitude which students resent [9].

DeMarco and Lister quote some interesting data from the University of New South Wales [10]. These data were derived from an annual survey which the authors did on programming projects in Australia. They were interested in motivational issues and how these affected programmers' productivity.

They found that when the programmer made the estimate for the required deadline for a job, productivity was measured at 8 (an index including various factors). When the supervisor made the estimate, productivity was 6.6, i.e. programmers didn't have ownership of the estimate, and hence didn't work as hard. When both programmer and supervisor made the estimate (jointly), productivity was 7.8, or almost as good as the programmer alone. However, when there was no estimate of the total effort required, productivity jumped to 12! These were situations where there was no fixed deadline, but a large amount of ownership.

This suggests that deadlines are often counterproductive (literally). If a deadline is necessary, it should be negotiated with the students. Even better, try no deadlines. Of course, the end of semester still looms large, but perhaps we need to work on that too. We just need to deal with the Administration Police!

DeMarco and Lister collect their own productivity data from what they call War Games. These are carefully supervised exercises where the task is well defined, and programmers work individually on tasks. The effort required is recorded in a log, and the resulting programs are tested for compliance against the specification. From many such cases, they have found some interesting statistics on personal productivity.

- The best person outperforms the worst by 10 to 1.
- The best is about 2.5 times the median performer.
- The upper half outperform the lower half by 2 to 1.

Assuming that learning is somewhat similar to building programs (both rather intellectual activities), then are we seriously neglecting the upper half of the class, and particularly the upper 10–20%? These figures suggest that those students could be doing outstanding work if we weren't throttling them!

Another response might be to say that the lower 25% of the class shouldn't be there. This may only be true of the type of material we are expecting them to learn. If our courses were more balanced towards the needs of real world engineers (in the author's case), and not academic engineers, then those students may well be excelling.

The physical environment

When employing people, it probably makes sense to try to make them comfortable, although it is obvious from DeMarco and Lister [7] that this is often not the case. An environment for intellectual work needs to be large enough and quiet enough for real thinking to occur. Noise and interruptions were frequently listed as distractions by programmers surveyed (p. 49).

This is important because to do intellectual work, we must be able to get ourselves into a state of concentration which DeMarco and Lister call flow—a state of deep, nearly meditative involvement (p. 63).

IBM did a study of some of their developers (reported on p. 62) which indicated that they typically spent 30% of their time working alone (presumably in this flow state), 50% working with one other person, and 20% with two or more people.

The difficulty with flow is that it takes time (15 minutes or more according to [7]) to reach this level of concentration. Noisy offices make this very difficult, which is why people often claim to need to go home or work after hours to get anything done.

For academics, it is the telephone and door knocking that makes flow so difficult. Assigning times when door knocking is not permitted and the telephone is redirected make good sense.

For students, our classrooms can make flow very difficult. Lectures (assuming little noise) are probably our best chance. Unfortunately, students often don't see it that way. They seem to only come for a set of notes. Tutorials, being less structured, have always been considered to be better learning environments; times when students can sit and think about questions raised in lectures. Alas, they are usually so noisy that flow cannot occur. This might explain why many students do not attend tutorials. They can't get any useful work done. They can't concentrate. Our computer laboratories are similarly noisy, particularly because formal classes have intruders present who add to the background noise. It is little wonder that many students do not attend.

Methodology and aptitude

DeMarco and Lister have a shot (chapter 17) at computing methodologies. In program design, methodologies have been proposed which provide rules for designing and implementing large projects (e.g. database systems). Many organisations adopt particular methodologies as standards. DeMarco and Lister point out that this can have an adverse effect on productivity because much of the fun goes out of the project; the programmer just works through the methodology.

It seems to the author that the way we teach engineering is like this. Our courses are full of methodologies and algorithms, for solving problems. Learning them is rather dull and boring. Curiously, when one has worked in an area for a while, it becomes obvious that there are, in fact, many other ways of tackling these same problems. If the learning situation could allow for this, then it is more interesting and educative.

Problem-based learning attempts to do just this, by not being restrained by a single lecturer's view of the discipline. There may well be some guiding notes to get students started, but after that point, they are expected to find solutions for themselves. This makes learning more interesting.

DeMarco and Lister describe aptitude tests often used by businesses to find suitable employees. Unfortunately, they state that most of these are left-brain oriented—based on logic. Further on in their careers, those hired will need to be right-brain oriented—creative, holistic, heuristic, intuitive. Aptitude tests may not be very useful in the long term.

Similarly most of what is covered in an engineering degree is left-brain stuff—mathematics and algorithms for solving standard problems. There has been some move to include communication and management, particularly hastened by the Institution of Engineers, Australia, but on closer examination, we teach these subjects in much the same way—standard solutions to standard problems.

Some efforts have been made to introduce more creative activities into engineering courses (e.g. [11]). It would be easy enough to require some right-brain activities in all subjects. Assignments can be extended in different ways. A problem-based learning approach favours such open-ended tasks, allowing groups to add their own flavour and ideas to the problem solution. This is important for developing creativity and initiative, but it is also important for developing ownership, which we have seen is very important.

Working in teams

In programming, as in problem-based learning, teams are vital. They provide the resources and the synergy for building large projects on the one hand, and for tackling significant learning tasks on the other. Cultivating teams is the topic of Part IV of DeMarco and Lister's book. They indicate that great teams are jelled teams. They have a strong sense of internal identity. Team interactions are everything, which is why methodologies (see earlier) don't work unless the team adopts them themselves.

What can we learn about learning? First of all, team learning seems to be the way to go, and we should encourage students to stay in teams for as long as it suits them—build on the sense of identity. Similarly, we should encourage those teams to come up with their own solutions and to publicise them for the benefit of everyone.

Members of teams should also not be working on more than one project—it is just too distracting. This is a real problem. Students have several subjects every semester. Perhaps they should spend two solid weeks on each one? Likewise staff are often working on many different things simultaneously. Is it any wonder that we find it so difficult to be productive?

DeMarco and Lister indicate in chapter 20 that one of the reasons that many managers are threatened by teams (and consciously squash them) is that management is not team oriented. They do not understand teams. Likewise, academics often work alone (although there has been a steady move towards specialist centres), and hence, to them, the idea of growing teams does not seem relevant.

DeMarco and Lister say that it is not possible to build teams only to create the right environment in which they might grow and flourish:

- make a cult of quality;
- provide lots of satisfying closure;
- build a sense of eliteness;
- allow and encourage heterogeneity;
- preserve and protect successful teams;
- provide strategic but not tactical direction.

The cult of quality has already been addressed. It was stated earlier that self-esteem is, to a large part, built on quality.

Closure is related to feedback and reinforcement. We all need some sense that what we are doing is worthwhile and (eventually) complete (in some sense). This is most obviously some form of assessment where the emphasis should be formative—congratulations on parts well done, and suggestions for improvement. Students should also be encouraged to improve the process through feedback to the lecturer.

Major universities (like Monash) pride themselves in being elite. Somehow that doesn't seem to translate to the students. Actively involving them in the department's activities could be a way of them identifying with the high standards to which the university itself aspires.

We should be delighted that we bring into our department a range of students with differing abilities. In the past, we have made the mistake that we try to arrange these people along a one dimensional scale of ability—some bright, some ordinary. We should have worked out by now that intelligence is a many dimensional attribute, and that if we design assignments appropriately, the person we thought was dull, has other talents previously unrecognised.

These assignments need to provide enough direction (strategic advice) without telling the students how to complete it. After all, that's where much of the satisfaction lies (see earlier discussion about methodologies). Similarly, any directions for extending the assignment should not stifle creativity.

Leadership

Managers or leaders? Managers are not usually members of programming teams just as academics are not part of student teams. Managers and academics can only create the environment that can make great things happen. Leadership is something else. It needs to come from within the team, and different people (students) will provide leadership at different times. If there is a permanent team leader, that person ceases to be a peer and teamwork begins to break down.

Academic departments could be effective teams, but the current method of appointing one or two professors to provide academic leadership flies in the face of DeMarco and Lister's advice. Heads of department need to see themselves as managers, and less as leaders. They create the environment for great things to happen. Leadership then comes from within the group as required.

ORGANISATIONS THAT EXCEL

DeMarco and Lister devote Chapter 16 to great companies—ones with low staff turnover that look after their people, where there is an energy and sense of belonging that is practically palpable (p. 112).

Does this happen in universities? Alas, probably not. Universities are certainly proud of their expertise and accomplishments, but they do not strive to be great organisations. Rather, they strive to encourage great individuals, the sum total of whom is great.

Within a university, it is only the academic staff who enter this summation. The non-academic staff are, on the whole, of little consequence, and the students, particularly undergraduates, are largely irrelevant. This is a sad loss to the summation. Is it at all surprisingly that after four years of neglect, only a small number of students remain for postgraduate study, or that the alumni have little devotion to the institution. (Their fondest memories are finishing.)

A great university should have a role for *all* its members—academics, researchers, non-academics, tenured and non-tenured, postgraduate and undergraduate students. There would be a shared vision and respect for each person's work, and its contribution to the total output of the organisation. This links neatly with Deming's ideas discussed earlier.

A plan of action

As teachers, we should be better **managers of learning**. This could happen in several ways.

- Make students and non-academic staff part of the department. They should be actively involved in all activities—including research.
- Create a suitable physical environment for individual learning. Although the library provides some private study space, space within the department is desirable. Students should have a strong sense of identity with the department as mentioned above.
- Provide space also for group learning. Most teaching rooms at the author's university are not suitable for this, but group learning is an important part of problem-based learning.
- Reconsider tutorial activities. It is likely that tutorials are not well suited to individual tasks. They are just too noisy and distracting. That type of activity needs to be done at home, or in separate study carrels. Tutorials should be built around group activities.
- Negotiate with students when assignments will be due and how to reward early submissions and penalise late ones. Ownership (by students) of this process is important.
- Emphasise quality and be prepared to be understanding both in terms of assignment content and in terms of assignment deadline, as mentioned above.
- Recognise that there may be a range of productivity of more than 2.5 times within the class.
- Design open-ended assignments. Specify the minimum requirements plus possible directions for extension. Have the (negotiated) marking scheme reflect this. It is likely (based on the author's experience) that more and more of the students will be working on the extensions. Such an approach develops creativity as well as providing scope for more productive students. Expect that students will find new solutions which you haven't thought of, and make sure that these ideas are made available to next year's class through written materials of some kind.
- Make assignments team-oriented, and encourage successful teams to stay together. Analyse team failures and turn them into constructive activities for future teamwork.

Many of these points are directly applicable to problem-based learning. This study of the management of software teams, an intellectual activity similar to learning, provides further corroboration for the team- and student-oriented approach of PBL.

Creativity

As the world becomes a more competitive place, there is a pressing need for organisations to be more creative, with the ability to generate new products, practices and processes to survive and prosper. Carr [12] addresses this need to be constantly creative. Similarly, we must develop educational processes which provide practice for students in creativity. As we have seen, problem-based learning can be used in this way through open-ended problem solving.

Carr states (p. 2) that creativity must be focused on the organisation's goals and objectives. Unless this is the case, creativity can be a distraction, drawing effort away from the organisation's main

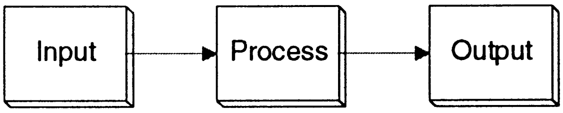

Fig. 1. Typical system module description.

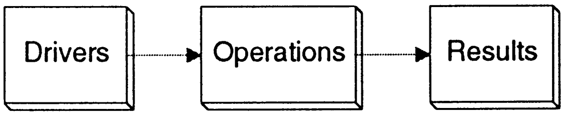

Fig. 2. Carr's revised systems model [12].

thrust. In a learning context creativity needs to be focused on the problem at hand. Unfocused creativity can be a waste of a student's time.

Carr provides four reasons why innovation is valuable:

- it's new—it leapfrogs current ideas;
- it fits the strategy (goals and objectives again);
- it creates value for both the company and the customers;
- it produces new practices, processes, products and/or services.

A **process** is defined as a mechanical series of steps while a **practice** is much more a social process even though there may be a fixed set of steps required. For example, a production line represents a process, while handling a loan to buy a car is a practice (it is much more people-oriented). Both processes and practices constitute *how* something is done. Innovation is possible here and Carr provides several examples of companies, e.g. Toyota, that excel by being innovative in how they work.

Products and services represent *what* a company is selling. There is an increasing move to services attached to products. For example, in the personal computer business, more value is to be gained from selling after-sales support than from the original sale of the hardware due to the low profit margins on hardware.

Creative qualities

Carr identifies seven core qualities of creative people:

- they intend to be creative;
- they direct the creativity toward goals they care about;
- they make high demands on themselves;
- they focus on important problems trying to find opportunities;
- they spend a significant amount of time formulating the problem before trying to solve it;
- consider many alternatives before committing themselves;
- they often make many attempts before coming up with a successful solution.

Hopefully, these are the types of qualities we expect from our students in their problem-based learning exercises—perhaps not in every task, but certainly across several tasks.

A SYSTEMS MODEL

Systems theory often proposes a simple Input-Process-Output model to describe complex systems such as work situations, Fig. 1 [12].

Carr recasts this using slightly different terminology (Fig. 2) where Drivers replace Inputs, Operations replace Processes, and Results replace Outputs. He further subdivides each of these into a more complex model (Fig. 3).

This new model allows us to focus on all of these components.

Resources

Consider first practices and processes at the centre of the diagram. They rely on resources to be successfully completed. Individual autonomy is very important if resources are to be successfully used. Further, a continual flow of resources is needed and if they are renewable, then the system can be productive in the long run [12, p. 29].

University departments are faced with a continual avalanche of information which must be processed and moved into new or revised courses and subjects. Traditionally, its resources have been staff, which are renewed at a slow rate determined by retirements and departures. It is unlikely that this is sustainable into the future as knowledge

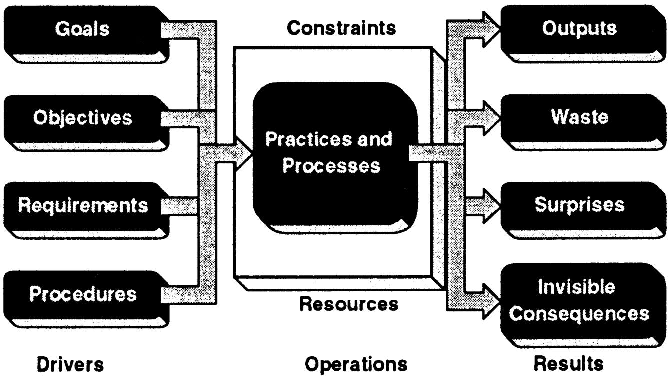

Fig. 3. The complete, revised systems model [12].

continues to grow exponentially. A resource we have which we have tended to neglect is our students. They can be more actively involved in sifting new information, building resource materials and teaching others.

Constraints

Practices and processes are also subject to constraints of various kinds. Oddly, these constraints most often come from the same people who provide the resources, and they limit access to the resources. Government bureaucracies are particularly good at this.

Teachers also unwittingly stray down this path. Assessment is probably the worst constraint we can apply to our students. We provide learning resources and situations, but then suggest an assessment regime which works counter to what we really wanted to achieve. As many who are experienced in problem-based learning know, the resources and constraints must be in the hands of those who must do the work (the students). They must be self-regulating [13]. Carr [12] puts it very succinctly:

- make those operating the system responsible for controlling it;
- then provide them with free access to the resources they need.

An organisation can then direct effort and resources away from control, and channel this energy into communicating goals, objectives and standards so that everyone in the organisation is pulling in the same direction.

Drivers

The goals, objectives, requirements and procedures from Fig. 3 represent successively more constrained statements of intent. For example, a **goal** might be 'to be Australia's International University', while a **procedure** might describe how to assess students studying on an allied campus in South East Asia.

We have already seen from the study of software teams that it is important that those responsible for the work, be closely involved in developing objectives, requirements and procedures. Carr also makes this point, and emphasises the need for mutually agreed objectives from the beginning.

In terms of our students, there is a hierarchy of such drivers. An individual staff member might be successful in developing a student-oriented learning environment in which students are involved in developing learning and assessment strategies. However, if the department as a whole is not committed to such an approach, students will not carry these new ideas to other subjects.

Results

Results come in many forms (Fig. 3). Outputs have been the traditional focus since that is our usual aim. A creative organisation and a creative teacher should be looking also at the other three components: waste, surprises and invisible consequences.

Creative organisations should be examining each of these three components for the gems which are often hidden there. For example, the weak adhesive developed by 3M was thought to be completely useless, but on creative reflection, it has spawned an entirely new product range: Post-It Notes.

How many times have you given your students open-ended problems and been delighted with the novel solutions? This is clearly an example of a creative organisation.

Many learning tasks are also sufficiently open-ended that the outputs are not clear and distinguishing outputs from waste is an important part of the learning. (Much research falls into this category). Students must have experience in this activity.

CREATIVE ORGANISATIONS

Creating a creative culture in our universities requires some pro-active steps.

- We must stress the importance of the seven attributes of creative people.
- We need renewable resources (students).
- The workers must be empowered to be self-regulating.
- Objectives must be mutually agreed upon.
- Waste must be identified (it is not always obvious) and opportunities identified in apparent waste or side-effects.

Problem-based learning is an effective means of achieving many of these points as long as learning tasks are not defined too narrowly.

Effecting change

Impediments exist in the attitudes and actions of the staff. Universities the world over have promulgated promotion strategies which encourage individual rather than team performance. This often works against the types of things discussed in this paper.

Carr [12] lists 10 attributes of a creative and flexible organisation.

- Build a high level of trust—this is the key ingredient.
- Tell it like it is (honesty).
- Communication should be unconstrained by departmental boundaries.
- Look for solutions, not scapegoats when mistakes occur.
- Focus on problems and opportunities, not on personalities and power—a tough one.
- Develop shared values, goals and objectives to support self-management.
- Include customers and suppliers.
- Anticipate change to create the future.

- Promote ownership.
- Encourage play and daydreaming.

Many of these ideas are common also to those of Deming [5] and DeMarco and Lister [7]. The key element is building trust. Unless we trust our students, we will not progress much further than our current state, and we will fail to gain the benefits that are waiting for us. Many academic staff treat students with contempt. (They often treat staff in other faculties and departments with contempt too, so this is not surprising). Turning this around into respect and trust will be a time consuming task. Deming and others state that a move to quality is at least a five-year project, so such changes cannot be made overnight.

Honest communication, based on trust, is then possible, allowing us to focus on the problems in our courses rather than be defensive about them.

We need to be creative in our use of self-management of students. The more trust we can provide, combined with self-management and self-assessment, the greater the rewards. The author has certainly been pleased with his self-directed, fourth-year computing elective [14]. Ownership of the learning, as encouraged by problem-based learning, is a powerful motivator.

We need to include our 'customers' and 'suppliers'. Customers include staff of all kinds, students, industry representatives, the professions and community and government representatives. Suppliers include schools and technical colleges. Carr reports that up to two thirds of innovations in some companies come from the customer/supplier interfaces. All too often we act defensively, justifying our current position rather than finding means to effect beneficial change for the future.

As we come to the end of this century, it is convenient to talk about the type of education our students might need in the new century and millennium. Apart from active student learning through styles such as problem-based learning, there seem to be few other changes happening in our universities which will build a new culture for the new century. This paper has attempted to sketch a new philosophy based on breaking down barriers between departments—between staff and students. This is a key element of Deming's philosophy.

SUMMARY

It is particularly pleasing to read the literature of working organisations, and to find it echoing the educational literature. It is further evidence that we are on the right track! In particular, all organisations are being encouraged to:

- empower workers so that they have the autonomy to improve their workplace;
- provide these employees with the resources so improvement can take place;
- build a culture of self-managing teams;
- develop creative employees;
- be flexible when laying down prescriptive guidelines for how work should be carried out;
- leadership qualities from all employees;
- provide a physical environment that is conducive to excellent work;
- develop a management style that encourages all these things (which is usually the first step).

When we look at active learning strategies like problem-based learning, we find many of these ideas already. However, the author believes that universities are still guilty of not implementing the ideas across all of their activities.

Any single department should be considering how it can make these principles *part of its culture*. Some of the principles have already been applied in research activities and with academic staff. However, we are well short of universal adoption, particularly in our treatment of undergraduate students. Once we stop thinking of undergraduates as recipients only of our wisdom, we will be able to recognise their contributions and draw them more fully into active roles in departmental activities such as research and community service. Only then will we become significantly more productive, and student learning will be far more interesting. It is a brave new world, appropriate for a new millennium!

REFERENCES

1. C. B. Adair-Heeley, *The Human Side of Just-in-time*, AMACOM (Amer. Mgt. Assoc.), New York (1991).
2. D. Boud and G. Feletti, *The Challenge of Problem-based Learning,* Kogan Page, London (1991).
3. G. Ryan (ed), *Research and Development in Problem-based Learning,* Vol. 1, PROBLARC, P.O. Box 555, Campbelltown, NSW, Australia (1993).
4. C. Engel, Not just a method but a way of learning, in Boud and Feletti [2].
5. W. E. Deming, *Out of the Crisis,* MIT Center for Advanced Engineering Study, Cambridge, Mass., (1986).
6. R. Hadgraft and D. Holecek, Towards total quality using problem-based learning, *Int. J. Engng. Educ.,* **11** (1) 1995, pp. 8–13.
7. T. DeMarco, and T. Lister, *Peopleware—Productive Projects and Teams,* Dorset House Publishing Co, New York, (1987).
8. P. B. Crosby, *Quality is Free: The Art of Making Quality Certain,* McGraw-Hill, New York, (1979).
9. A. P. Prosser, Doing something about student motivation, *HERDSA News,* **17** (2), July 1995.

10. M. Lawrence and D. R. Jeffrey, Commercial programming productivity—an empirical look at Intuition, *Australian Computer Journal,* **15** (1), 1983, p. 28.
11. B. W. Field, Do undergraduate engineering courses suppress creativity? *Proc. 5th AaeE Conference on Engineering Education,* Univ. of Auckland, NZ, Dec 1993, pp. 536–541.
12. C. Carr, *The Competitive Power of Constant Creativity,* AMACOM (Amer. Mgt. Assoc.), New York (1994).
13. D. R. Woods, *Problem-based learning: How to gain the most from PBL,* Donald R. Woods, McMaster University, Canada (1994).
14. R. G. Hadgraft, Experience with a problem-based, fourth year computing elective, in *Civil Engineering, 7th Asian Assoc for Eng Educ Conf,* Melbourne University, Dec 1995, pp. 40–44.

Roger Hadgraft is a senior lecturer in the Department of Civil Engineering at Monash University, with special responsibilities for computing within the curriculum and the department in general. He has BE and M.Eng.Sc. degrees in Civil Engineering from James Cook University, Townsville, Australia, including a University Medal, and a Dip.Comp.Sc. from the University of Queensland. He has worked for the Queensland Water Resources Commission in a variety of roles, including a significant amount of innovative computer programming. For the last 10 years, Roger has been teaching at Monash. His special research interests are in the effective use of computer technology both for teaching purposes and in engineering practice. In particular, he has been exploring the use of hypertext to deliver undergraduate course materials as well as information for practising engineers, and has developed a software tool (called *DR Help*) to make this task much simpler. He has recently combined his interests in problem-based learning with his hypertext course materials to suggest a resource and problem-based approach for engineering education in the future. He was awarded a Faculty Teaching Award in 1991. In the last few years he has been exploring quality issues, and how these relate to the types of student-oriented learning environments he has been creating through on-line course notes and problem-based learning. Recently he has been writing on the benefits of extending quality principles to staff, as well as to students.