

# Lifelong Learning for Software Engineers

**Dr Lorna Uden**

School of Computing  
Staffordshire University  
The Octagon, Beaconside  
Stafford, ST18 0AT, UK  
l.uden@staffs.ac.uk

**Alan Dix**

aQtive limited and Lancaster University  
Birmingham Research Park  
Vincent Drive, Birmingham, B15 2SQ, UK  
alan@hcibook.com  
<http://www.hiraeth.com/alan/>

## Abstract

Software engineers need problem-solving, critical thinking and metacognitive skills in addition to their technical expertise. They must have lifelong learning skills to cope with the ever-changing nature of the software evolution. This paper describes case studies of how these lifelong learning skills can be acquired through Problem-based learning. Due to increasingly limited resources in higher education, students must adopt more independent and learner-centred approaches. In order to address the needs of such students, a final year computing science module was developed in which students working in groups learned how to learn through problem-based learning (PBL). Studies over the two years of running the ISD module revealed that students who took the module became better students, able to transfer their learning to other modules and work. Feedback from graduates now in employment comment on how the module has enabled them to cope better with their software engineering jobs.

## Keywords

Problem-based learning, software engineering education, lifelong learning.

## 1. Introduction

In a dynamic technical environment it is no longer sufficient simply to teach Software Engineering students about their subject: the content and context will have changed almost before they reach their first job.

This issue has been recognised world-wide and over many disciplines. A recent report on education in USA, concluded that society's future depends on a citizenry that can think and reason creatively and deliberately; develop sound judgements of information (and) understand and contend effectively with rapid and constant change [8].

Similar conclusions were reached by an Australian government report, "the major function of education is .... to increase the individual's capacity to learn, to provide students with a framework with which to analyse problems and to increase their capacity to deal with new information.". [1]. In other words, the purpose of education should include not only the acquisition of knowledge and skills, but also the development and improvement of higher order thinking strategies.

In the UK, the Dearing Report [5] recommended all staff at institutes of higher education give high priority to developing and implementing learning and teaching strategies which focus on the promotion of students' learning.

These various reports are addressed at general education, but software engineering is particularly problematic due to the rapid rate of technological change. In 1989, software trainers in industry and the military from North America and Europe met as a working group to share experiences on the practice of software engineering training. Among training aspects of concern, the working group focused on the transfer of skills, perhaps the most significant issue for workers in an environment where high technology one year can become kids toys in 5 years time. Transfer has taken place when trainees can be observed to do what they were trained to do, using the resources they need where and when they need them to perform job tasks at a satisfactory level [4].

One of the inputs to this workshop was a survey conducted by the Software Engineering Institute Technology Transition Program to establish a general picture of the then current state of training in software engineering in industry and the military. Of the total number of respondents (47 software training professionals nation-wide) only 29% considered that software engineering training (SET) transferred even moderately well to job situations. This result is disturbing considering that the ultimate goal of

education is the transfer of knowledge and skills to the solution of 'real-life' problems. Although this snapshot of SE education was taken in 1989, the situation 10 years on is probably not significantly better.

Employers expect that software engineers will continue learning throughout their professional lives to meet the rapid changes of technological innovation and keep abreast with the latest knowledge and skills in software engineering. This requires the learning of problem-solving, critical thinking, metacognitive and self-directed learning skills, rather than specific skills and knowledge. This change in expectations of employers has important implications for the teaching and learning of our software engineering undergraduates.

We have been concerned about the frustration faced by software engineering students in their learning. Many students have experienced that learning is difficult, especially when it comes to problem-solving in software engineering. They have no idea where to begin, despite their familiarity with the syntax of programming languages. These students can memorise facts and procedures, but are unable to explain observed phenomena, to solve real-world problems, or to analyse problems and to think critically. Many of these students may pass examinations, but be unable to *apply* the same knowledge to solve new problems. They have difficulty in utilising the knowledge and skills acquired via formal learning. It is clear from contemporary thought in the field of education that lecture style methods of teaching are not effective. We must change the way we teach and help students learn to solve problems and think independently [3]. That is to develop strategies that teach content in ways that also teach thinking and problem-solving skills.

These higher-level skills are termed metacognitive skills. This refers to knowledge and awareness of one's own cognitive processes and the ability to actively control and manage those processes. Considerable evidence has accumulated that suggests an emphasis on metacognition during training can result in significant improvements in problem-solving for the tasks, as well as in transfer of skills across tasks [7].

In order to help students to acquire these important skills, problem-based learning was adopted in one of our software engineering modules. Instructional Software Design (ISD) is a final year computing science degree module option at Staffordshire University in the UK. The module has dual aims.

First students learn about theories of learning and instructional design methods in order to be able to construct effective instructional software. In addition, students are taught using PBL techniques that they can then apply to their own learning in this course, in other courses, and throughout their future careers. It is the latter which gives students transferable skills such as problem-solving, critical thinking, metacognitive and independent learning skills. In this paper we will describe and evaluate the experience of running this module. In particular we will see how the student's learning styles were affected both during and after their course.

In the next section we will give a brief review of PBL and why it is of particular value in software engineering. The next section looks at the particular tutorial processes we employed. Section Four describes the benefits of the approach, both the immediate and longer-term. Our experience and the lessons learned will be the focus of Section Five and, finally, Section Six gives a summary of the paper.

## 2. Problem-Based Learning

### What is PBL?

Problem-based learning (PBL), according to Barrows [2] is, "... the learning which results from the process of working towards the understanding of, or resolution of, a problem." PBL is a way of constructing and teaching courses using problems as the stimulus and focus for student activities. The courses start with problems rather than the exposition of disciplinary knowledge. They move the students towards the acquisition of knowledge and skills through a staged sequence of problems presented in context, together with associated learning materials and support from teachers.

Barrows describes the main educational goals as:

- To develop students' thinking or reasoning skills (problem solving, meta-cognition, critical thinking) and;
- To help the students become independent, self-directed learners (learning to learn, learning management).

The purpose of PBL is to produce students who will:

- Engage a challenge (problem, complex task, and situation) with initiative and enthusiasm;
- Reason effectively, accurately, and creatively from an integrated, flexible, usable knowledge base;
- Monitor and assess their own adequacy to achieve a desirable outcome given a challenge;

	<b>Activity</b>	<b>By Whom</b>
Phase 1	<ul style="list-style-type: none"> <li>• present problem</li> <li>• set hypothesis</li> <li>• identify learning</li> <li>• assign tasks</li> </ul>	<ul style="list-style-type: none"> <li>• tutor</li> <li>• group (with tutor)</li> <li>• group (with tutor)</li> <li>• group (with tutor)</li> </ul>
Phase 2	<ul style="list-style-type: none"> <li>• consult resources</li> <li>• detailed study of issues</li> </ul>	<ul style="list-style-type: none"> <li>• individual and group</li> <li>• individual</li> </ul>
Phase 3	<ul style="list-style-type: none"> <li>• evaluate resources</li> <li>• re-examine the problem</li> <li>• revise hypothesis and learning issues</li> </ul>	<ul style="list-style-type: none"> <li>• group (with tutor)</li> <li>• group (with tutor)</li> <li>• group (with tutor)</li> </ul>
Phase 4	<ul style="list-style-type: none"> <li>• present findings</li> <li>• metacognitive critique</li> <li>• reformulate further hypotheses</li> </ul>	<ul style="list-style-type: none"> <li>• group to peers and tutor</li> <li>• tutor and peers</li> <li>• group (with tutor)</li> </ul>

Figure 1: The Four Phases of PBL

- Address their own perceived inadequacies in knowledge and skills effectively and efficiently;
- Collaborate effectively as a member of a team working to achieve a common goal.

PBL is a challenging and motivating way to learn because students take ownership of their problem and work in real-world situations. They perceive learning as important and relevant to their own lives.

PBL is centred on the learning that emanates from a real problem. In PBL, students spend time in learning – by identifying what they need to know, by finding out, by talking to each other and by applying their new knowledge. The primary aim is learning itself not the completion of the project – the project is the means to the end. Note that this is different from standard project work in that the ways in which the students are encouraged to tackle the problem are designed to encourage learning in a structured manner (albeit with substantial learner control). It is also different from apprenticeship and 'learning on the job' where the focus is on completing the work and learning as a by product.

In short, the key ingredients of PBL are:

- the problem as the focus of learning
- learning as the purpose of the problem
- the problem as the integrator of concepts and skills
- commitment to self-learning

In addition, PBL is typically used in teamwork and small group situations as this encourages the development of reflective abilities. This can be achieved individually, but is often easier in a group situation.

### **PBL and SE**

The general model for problem-based learning was developed in the field of medical education in the 1950's. Since then PBL has been applied successfully in teaching many subjects. Among these are medicine, business, law, engineering, agriculture, architecture, management, social sciences, etc. PBL is particularly relevant for software engineers who are faced with a constantly faced with a changing technical (and increasingly business) environment. They need to be able to learn throughout their working life, both independently and as part of learning teams.

PBL encourages independent learning and deeper understanding of the material rather than superficial coverage, giving students practice in tackling engaging problems and defining their own gaps in understanding of the context of those problems.

The small group setting used in PBL encourages an inquisitive and detailed look at all issues, concepts and principles contained within the problem. The time spent outside of the group facilitates the development of skills such as literature retrieval, critical appraisal of available information and the seeking of opinions of peers and specialists. Thus PBL encourages the students to become responsible for their own and their colleagues learning.

In short, PBL provides students with the opportunity to develop many skills including: appreciating the diversity of inputs from different group members, time management, information retrieval, communication, and self-managed learning; all critical for the practicing software engineer.

IDEA (hypothesis)	Facts	Learning issues	Action Plan
Students' conjectures regarding the problem. – may involve causation, effect, possible resolution, etc.	A growing synthesis of information abstracted through inquiry, important to the hypothesis to be generated	Students' list of what they need to know or understand in order to complete the problem task.	Things that need to be done in order to complete the problem task.

Figure 2: PBL Tutorial Chart

### 3. The PBL Tutorial Process

There are many strategies for implementing PBL. The particular PBL model adopted was that of Barrows [2]. This section briefly reviews the PBL tutorial process we used. It occurs in four phases, as summarized in Figure 1.

In the ISD module PBL was used as a tutorial process. As part of the assessment for the module, students have to complete two research projects working in-groups. An example of a given problem is, "Develop a hypermedia courseware for learning applied optics using the WWW".

The tutor presents the problem 'cold' to the students who do not know what the problem will be until it is presented. The students discuss the problem, generate hypotheses based on experience they have, identify relevant facts in the case, and learning issues. The learning issues are topics of any sort deemed of potential relevance to the problem and which the group members' feel they do not understand as well as they should.

To help students to structure their thought processes, a four-column chart shown in Figure 2 is used (adapted from Duffy [6]). A session is not completed until each student has an opportunity to verbally reflect on their beliefs about the problem, and assume responsibility for particular learning issues that were identified.

#### Starting a new problem

When the students have gone as far as they can with the problem, they determine what resources they will use (faculty experts, library, Internet, etc.) to gain the knowledge and skills needed. They also assign learning issues to different members of the group to work on. A time limit would also be set for the completion of the task. After the session, the students all engage in self-directed learning where they work independently of the tutor consult resources and work collaboratively. The student group, depending on the extent and depth of issues they have elected to pursue negotiates the length of this phase.

After self-directed learning, the students meet again.

They apply the newly gained knowledge back to the problem, critique their prior thinking and knowledge, and refine their understanding of the problem and its management. They then synthesize what they have learned, relate it to prior problems and anticipate how it might help with future problems. They also evaluate resources - what is most useful and what is not so useful. This cycle may repeat itself if new learning issues arise. The students also assess themselves individually in the following areas: problem solving skills, knowledge acquisition, self-directed learning and support of the group. The last phase is the presentation of the solution by groups to peers and the tutor.

While self-directed learning is an important element, PBL is not an independent study curriculum. Each student works as a member of the tutorial group, and the group works together in resolving the problem. As a result, teamwork is an essential ingredient in PBL.

During the tutorial process, the tutor guides the students in reasoning their way through the problem. Significant findings are recorded by the group, along with their hypotheses and learning issues, knowledge needed to better understand and further pursue the problem. Using tutorial skills, the tutor facilitates student access to their own prior knowledge as well as the identification of the limitations of their knowledge. The tutor also guides students to articulate their knowledge of the relevant disciplines as they relate to the problem at hand. As students progress through the curriculum, they learn to reason through the problem effectively and efficiently. The need for information required understanding the problem generates learning issues for further study.

For the learning issues identified, an action plan, consisting of a list of activities that students need to do to achieve the learning issues, is worked out. The action plan lists the types of resources, which are needed to solve the problem. Resources may be books, journals, the Internet, etc. If the Internet is involved, students must work out exactly what they want to look for, based on the learning issues

identified. In fact, in the case of ISD, the Internet was a crucial resource for most of the chosen problems. Students found the tutorial process very useful because they were able to identify the relevant information to search for, rather than browsing through materials they did not want or need.

## 4. Benefits of PBL

### Immediate Outcomes

Commonly stated benefits of PBL include, increased retention of data; integration of knowledge; life-long learning; motivation to learn; development of reasoning and critical thinking skills; development of communication and interpersonal skills; and development of the ability to work effectively in a team.

Evaluating PBL is difficult because many of those benefits only become evident over a student's lifetime. However, a post-module evaluation questionnaire revealed that students:

- Enjoyed the opportunity to learn things for themselves (90%);
- Felt that they had learned from each other (85%);
- Felt that PBL gave greater confidence in tackling problems themselves (90%);
- Felt that PBL helped them to develop their thinking and problem solving skills (95%);
- Developed project management skills (80%).

Of course, achieving those results required hard work for both the tutor and the students. The questionnaire results showed that over 98% of the students rated PBL as the best part of the course. In addition, students reported that they applied the techniques learnt in this module to other parts of their course. One student wrote, "When applied in other course modules which I have learnt, it helps me to remember what I have done because I know why I do it and how I can apply it when given a problem". Another student reported that he now applied PBL methods to all aspects of his life including managing his personal finances!

### Long-term benefits

Now that the first cohort of students in our case study have graduated and found jobs in the 'real world' we are continually receiving feedback as to how much the techniques outlined in this paper have helped them in real-life situations. Below are selections of the comments received from graduates who are currently working:

**William Toh** from Singapore wrote to say that he used PBL to apply for his jobs. In his current job as a SAP Consultant he says,

*"Confronted with limited information, I use the various problem-solving techniques taught in ISD to provide consultation for my clients."*

**David Grocott** wrote from BT (UK):

*"The ISD module was very useful to me ... in order to build our two working prototypes in time..."*

From **Doris Ong** and **Te-Ping Wong** (both working as software engineers at National Computer Systems):

*"As software engineers, ISD has equipped us to be better team players, ... to master new languages and software tools."*

**Jason Gass** from Digital Equipment UK writes:

*"I personally found the ISD module extremely useful, particularly the principles of Problem-Based Learning, which I still use to tackle day-to-day research problems."*

Of course, this is only anecdotal evidence. However, it is rare to find graduates who are able to point to specific elements in their course and relate them to their job. This makes the above comments more substantial as an assessment of the course success. Even more important is the fact that the students felt able to make this an assessment - that is, they are clearly demonstrating critical evaluation of their own learning - a metacognitive skill that goes beyond the content of the specific module.

## 5. Lessons Learned

We have applied PBL to ISD over two years now and have learned many lessons. There were times of encouragement and frustration too, especially early in the course when students were facing a major change in their educational mindset, when motivation and encouragement were important. Above all we have learned that PBL is not easy!

### Domain Analysis

In designing a PBL curriculum, we have to analyse what must be learned. In doing this we have to combine identification of key concepts, procedures, etc., with analysis of the professional use of those concepts. Identification of key concepts is a matter of what is most important for students to "know". This must include the learning outcomes of the module as stated in the degree scheme. The identification of key

concepts is defined through the professional activity that calls for their use; that is, it is defined in the activity of the learner. In our case, the activity was to do with designing instructional software to solve user's problems.

## **Types of Learning**

There are two things that must be emphasised. First our analysis must not preclude any type of learning activity - memorisation of a list, or extensive practice of a skill may be necessary - but it should arise out of the need to use the information in authentic tasks. Second, what must be learned includes not only information in the content domain but also metacognitive, collaborative and other skills which are necessary for participating in authentic activity. In particular, a crucial issue is how to ensure that the problems designed are educationally rich enough that in seeking answers students must gain understanding of significant subject matter concepts. Problems also need to be feasible and manageable given the time and resources available to the students. However, artifacts should be rich enough to promote both depth and breadth of knowledge in their creation as well as demonstrate student mastery of the content. Artifacts must require students to integrate information and use complex thought.

Problems must be designed to sustain student motivation and thoughtfulness. A number of factors should be considered in problem design that affect whether students will be motivated to do the problem in a manner that fosters understanding. These factors include whether students find the problems to be interesting and valuable, whether they perceive that they have the competence to engage in and complete the problem, and whether they focus on learning rather than outcomes and grades. Initially the students found PBL very difficult and frustrating as they were so used to passive learning. They frequently asked for answers, which we were tempted to give, but our role was one of facilitators of thought in the students, not of answer providers. Patience was required on both sides as we attempted students to think for themselves. Students initially lack proficiency in cognitive and metacognitive skills to generate plans, systematically make test predictions, interpret evidence in the light of those predictions, and determine solutions. One of the techniques which they found particularly helpful was 'thinking aloud', that is verbalising their thought processes as they attempted to think through a problem. This required us to explain and demonstrate the technique quite early in the PBL sessions. We found that the interest

and value students attributed to the problem affected how motivated they would be to engage in their learning. Students interest and perceived value are enhanced when:

- the tasks are varied and include novel elements;
- the problem is authentic and has value;
- the problem is challenging;
- there is closure, so that an artifact is created;
- there is choice about what and/or how work is done;
- there is the opportunity to work with others.

## **Changing Attitudes to Learning**

Another frustration experienced by the students was their initial inability to integrate the diverse ideas generated by the nature of the problems they were investigating. It was necessary to constantly reassure and motivate them without actually providing the answers. By challenging their guesses and assumptions we were gradually able to focus their minds on relevant factors and enable them to reject ideas which lead only into "blind alleys". All of this was very labour intensive, but after a few weeks perseverance the students began to realise what a powerful tool we were putting into their hands. Once the light dawned, the students began to use PBL on other areas of their study, and reported back how much the concept had enabled and empowered their critical thinking skills, not only on the initial problem set, but in all aspects of their study. Students often came with different attitudes from ours towards their learning. Many perceived learning as a means to external ends such as grades and status among their peers and tutor, and above all obtaining good examination grades. Because of this, they tended to focus on low level learning without having the need to be involved in higher order cognitively processing. We had to work hard to change these attitudes.

## **Collaborative Learning**

It is very important that a software engineer has the ability to work effectively as a member of a team. Although students had to work in groups, most of them preferred to work as individuals. This is because very often in groupwork, one member of the group works whilst the others do nothing, but still pick up the marks. Thus, initially, students were very reluctant to work collaboratively. Furthermore, many of the students did not have the skills to benefit from collaborative work. Working with others requires that students are able to discuss ideas, communicate

clearly, consider alternatives systematically, monitor their own understanding, compare their points of view with those of others, ask clear questions and share the workload fairly.

As it was a requirement that the students work collaboratively, we had to instill into these students the concept of a team that operated as one unit rather than 'groupwork' where one or two people do all the work and the rest observe. This was not easy for students or for ourselves and required us to have regular sessions with the students to ensure true teamwork resulted. They soon learned how to discuss issues, share and plan work within the group. Their attitudes toward collaborative work changed as the PBL session progressed. Students began to see the benefits of sharing work and working as a team to solve complex tasks. They also began to enjoy each other's company and fostered social as well as academic relationships.

The evidence of team work made such an impact that they sought to work with other students outside of the university. In the main, this was achieved by e-mail and computer-mediated conferencing facilities. A CMC discussion group was set up via the Internet between postgraduate students in computer-based instructional design at a university in the USA and our own students. The discussion centred on current issues, strategies and methodologies for computer-based design and development, forming into teams and designing computer-based instructional products, and making designs available to other teams for comment and critique. The unique part of this discussion framework was that the students would be discussing issues with *everyone*, including members of the class in the USA, and would be receiving comment and criticism from the other class as well. The CMC discussion was remarkably successful and both groups of students enjoyed learning from each other.

### **Learning can be great fun**

We feel strongly that students should value learning as an end in itself. To achieve this we want them to see learning as fun in order to benefit from assignments and demonstrate greater levels of cognitive engagement in their work, be able to apply critical thinking, metacognitive and self-directed learning skills to the development of understanding of subject content. This was perhaps one of the hardest tasks we faced. However, as we continued the PBL sessions, the students began to see the benefits of being cognitively engaged in their work. As they became more and more involved with the problems,

they began to really enjoy the higher level of processing demanded of them. Many had a change of mindset. These students found learning so interesting that one of them actually openly expressed that he did not realise learning is such fun.

## **6. Summary**

There is no question that software engineers must have a rich and extensive knowledge base in order to practice their professional skills. However knowledge is of no value if the software engineer cannot recall or apply it to solve problems. Conventional software engineering curricula often do not explicitly stress the development of problem solving, critical thinking and metacognitive skills. In a rapidly changing world, we need to train our future software engineers to have skills that equip them for lifelong learning. The case study has shown us that Problem-Based Learning has important implications for the teaching and learning of software engineering.

## **References**

- 1 Australian Government Department of Employment, Education and Training, (1987) *Policy Discussion Paper on Higher education*.
- 2 Barrows, H.S. *The tutorial process*, Southern Illinois University School of Medicine, Springfield, IL., (1992).
- 3 Bransford, J.D., Sherwood, R.D., Hasselbring, T.S., Kinzere, C.K. & Williams, S.M. (1990). Anchored Instruction: why we need it and how technology can help. In D. Hix & R. Spiro (eds), *Cognition, Education and Multimedia: Exploring Ideas in High Technology*. pp. 115-141. Lawrence Erlbaum., Hillside, NJ.
- 4 Delgado, "Issues in transfer of skills in software engineering training", *ACM Software Engineering Notes*, Vol. 15, No.2, 1990, pp 23-25.
- 5 DfEE. *Higher Education in the Learning Society – Report of the National Committee of Inquiry into Higher Education*, R. Dearing, (chair). Department for Education and Employment (1997) Available at <<http://www.leeds.ac.uk/ncihe/>>
- 6 Duffy, T. (1994). Problem-based learning. *NATO Advanced Studies Institute, Supporting Learning in Computer Environments*, Herriot-Watt University, Edinburgh, Scotland, (August 1994).
- 7 McCrindle, C. & Christensen, (1995). The impact of learning journals on metacognitive and cognitive processes and learning performance, *Learning and Instruction*, 5, 167-185.

