# Steps in developing an advanced software engineering course using problem based learning

Lyn Brodie, Hong Zhou and Anthony Gibbons

## Abstract

**University graduates may struggle to convert the skills they have learnt in software engineering design principles to real-world situations such as would be found in industry. The traditional teaching practice of lectures and tutorials is not providing the context nor sufficient practice for students to develop the skills needed to solve real work problems. This paper investigates the use of Problem Based Learning (PBL) and its application to software engineering and distance education. Through a sound pedagogical approach the key skills of PBL (as endorsed by PBL practitioners, such as problem solving and independent learning) can be developed in the students as they are exposed to real world software engineering problems.**

## Introduction

Software systems are often complex and shifting, and the development of such systems can be extremely challenging. Equipping software engineering graduates with the skills and abilities needed by the industry to meet these challenges is very important to an institution of education.

Software engineering is a developing area of study within the programme offered by the Faculty of Engineering and Surveying at the University of Southern Queensland (USQ). USQ has a unique study environment and culture, and developing a curriculum to suit our student cohort, as well as industry standards, challenges academics to innovate and improve study material and delivery methods.

### Background to USQ and its PBL courses

USQ is a regional university, with its main campus in Toowoomba located approximately 150 km inland from the state capital, Brisbane. The University also operates two other campuses in Queensland at Springfield and Fraser Coast. The University began operation in 1967 as an Institute of Advanced Education and gained university status in 1990. During this time it has gained an international reputation for distance and flexible education. In addition to face to face classes, the University offers the majority of its courses in traditional print, online and enhanced modes for study by distance students. Indeed the majority (75%) of the University's 26,000 students study away from the campuses using these flexible delivery modes. Only 25% of students enrolled at the University study in a traditional campus setting.

The University has five faculties including Engineering and Surveying (FoES). FoES offers a variety of undergraduate and postgraduate qualifications. The undergraduate programmes include: a two year Associate Degree, a three year Bachelor of Technology, a four year Bachelor's Degree and five year Double Degree programmes. These programmes are fully articulated, offering flexible entry points and catering for students of diverse educational backgrounds, as shown in Figure 1. The Faculty is unique in that there are no departmental divisions. Many of the large core courses, particularly in the first year, are team-taught by staff from all disciplines.

In 2001 the Faculty introduced a strand of four core courses to teach fundamental technical knowledge and key generic attributes. These courses use a Problem Based Learning (PBL) methodology which was considered by the Faculty to be best suited to deliver the graduate attributes now required by the engineering accreditation body, Engineers Australia. Accreditation worldwide is now focusing on graduate outcomes rather than the educational process. There is an increasing emphasis on teamwork, communication skills (oral and written, formal and informal), problem solving skills and lifelong learning (IEEE, 1996; IEAUST, 1999; ABET, 2000; Engineering Council UK (ECUK), 2003). PBL can deliver these required attributes whilst also exposing students to complex real world engineering problems relevant to their future professional careers (Felder et al., 2000).

The four PBL courses are delivered to all students of the Faculty, both on-campus and distance. The students work in multidisciplinary teams solving complex real world engineering problems. The distance students utilise a variety of electronic communication methods, both synchronous and asynchronous, working in teams to solve the given problems and meet individual learning goals which are aligned with particular course specifications (Brodie, 2006; Brodie, 2007a).

These four courses have been very successful and have been recognised in both University and national teaching and curriculum awards. They deliver significant benefits to students, staff and the Faculty including, for students, increased retention and pass rates, extension of existing skills, recognition of prior knowledge and independent learning and research skills. The courses are routinely used as training for staff to gain experience in cooperative learning techniques which assist in updating didactic teaching methods. The Faculty has also formed a Centre for Engineering Education Research to support staff in scholarly and research activities centred on learning and teaching.

It has been demonstrated that the difficulties of delivering PBL to distance students dispersed around the world and working in different time zones (such as completing team projects and meeting individual learning goals) can be overcome by using an appropriate Learning Management System (LMS). This provides team and individual facilitation to overcome team and technical problems and develops a community of practice within and between teams (Brodie and Gibbings, 2007).

## Background to PBL

Most current literature is based around the introduction of PBL to the medical school at McMaster University in Canada in the 1960s, but its intellectual history is much older. Thomas Corts of Samford University describes PBL as ‘a newly recovered style of learning’ (Rhem, 1998).

PBL is a pedagogical strategy where students are presented with open-ended, contextualised, real world situations. They develop content knowledge, application of knowledge and problem-solving skills by defining the problem, sourcing resources (including prior knowledge and experience of team members) and identifying gaps in their own knowledge (Mayo et al., 1993; Brodie, 2007b). PBL is now a widespread teaching method in disciplines where students must learn to apply knowledge,

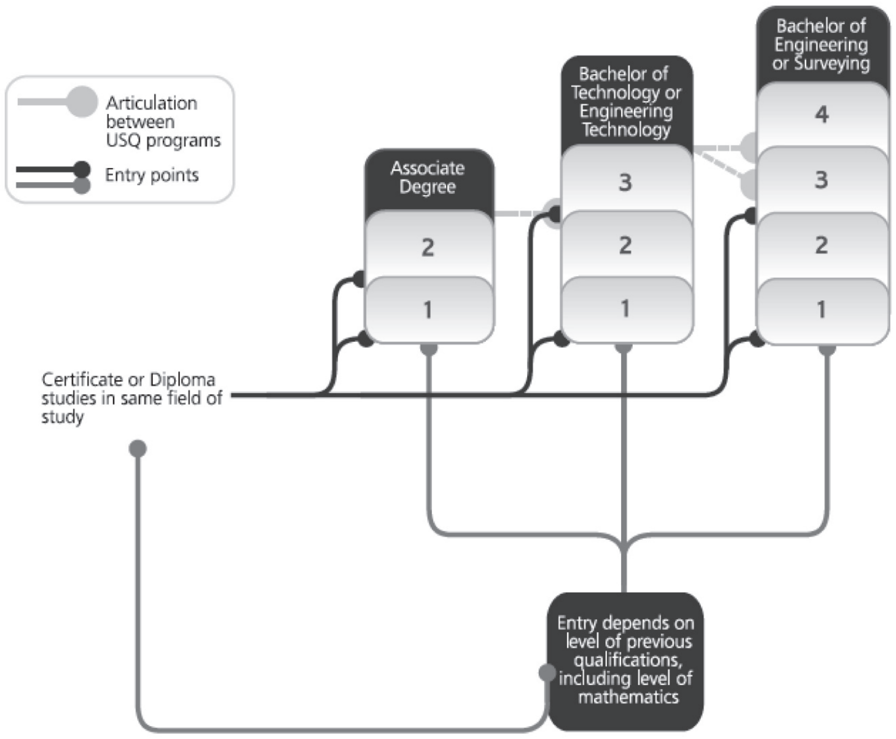
Articulation between USQ programs
Entry points
Associate Degree
2
1
Bachelor of Technology or Engineering Technology
3
2
1
Bachelor of Engineering or Surveying
4
3
2
1
Certificate or Diploma studies in same field of study
Entry depends on level of previous qualifications, including level of mathematics

**Figure 1.**
Articulation of Faculty programmes (http://www.usq.edu.au/resources/ugengineeringweb.pdf)

not just acquire it (Wilkerson and Gijselaers, 1996; Brodeur et al., 2002).

Student learning occurs within small group discussions and the academic assumes the role of a facilitator, not a lecturer (Aspy et al., 1993; Barrows, 2000). Thus the amount of direct instruction is reduced and students assume a greater responsibility for their own learning (Bridges and Hallinger, 1992). As they can share prior knowledge and experience with the group, mentoring and peer assistance assumes a more prominent role in the student learning experience and helps build a learning community. This shared and interdependent learning experience can be successfully carried out in an online or virtual environment, given appropriate scaffolding. The novel approach taken by the Faculty in delivering PBL to distance education students supports learning in virtual teams and develops problem-solving skills (Brodie and Gibbings, 2007; Gibbings and Brodie (in press)).

The educational and philosophical theories underpinning PBL were not explicit in early PBL literature (Newman et al., 2001; Rideout and Carpio, 2001) and the pioneers of the McMaster programme had no background in either education or psychology. They simply thought that learning in small teams using authentic cases and problems would make medical education more interesting and relevant to their students (Barrows, 2000; Newman et al., 2001). This current PBL methodology is now used in more than 80% of medical schools in the USA (Vernon and Blake, 1993).

From these beginnings, PBL has been incorporated into a wide range of professional studies including nursing, dentistry, social work, management, engineering and architecture (Boud and Feletti, 1997) and spawned a plethora of educational terminologies with an almost unclassifiable array of categories (Barrows, 2000).

In this paper, PBL is defined as:

*[…] a constructivist learning paradigm where small groups of students, engage in cooperative learning and collaborative problem-solving to solve problems in complex and authentic projects. These projects pursue specified learning outcomes that are in line with academic standards and course objectives with assessment focusing, to a varying degree, on the project outcome versus team process. (Brodie and Borch, 2004)*

## PBL and distance education

PBL has been slow to be integrated into distance education. There are limited references to PBL in distance education and all of the documented situations rely, at least in part, on some face to face interaction between the students. In many situations, distance PBL simply refers to students meeting off the main campus of a university, but still meeting face to face.

For application to the Faculty's situation, any course utilising PBL must be able to deliver the same benefits to teams working virtually as for those teams working face to face. Research conducted by the PBL teaching team at USQ shows that even the on-campus students are now fully embracing the electronic discussion methods and virtual team work primarily set up for distance students (Brodie, 2006; Brodie and Gibbings, 2007). The student cohort develops a 'community of practice' where students interact not only to solve the given problems but also on a social level. Mentoring and peer assistance is strongly encouraged, and in fact rewarded by the assessment scheme (Gibbings and Brodie, 2008).

The PBL courses at USQ utilise a Learning Management System (LMS) – WebCT Vista 4 –which is the standard LMS for the University. This is delivered through a student "StudyDesk" and provides discussion boards and chat facilities for each team, electronic submission of both individual and team work and the ability to deliver online assessments and surveys. Any future courses using PBL can make use of a significant body of scholarship, research and experience developed in the Faculty in delivering PBL to distance students (Helbo et al., 2001; Aravinthan and Worden, 2006; Brodie et al., 2006; Brodie and Gibbings, 2007).

## Background to software engineering

Software engineering is an engineering discipline which is concerned with all aspects of software production, from the early stages of system specification through to maintaining the system after it has gone into use (Sommerville, 2007). Many people see software development as something done by individuals spending long hours in front of a computer. Hogan and Thomas (2005) believe this image is reinforced by secondary and tertiary education where students must complete simple programming tasks entirely on their own. Indeed it is common to *punish* group collaboration. However, the professional practice is dominated by the team environment and thus individual programming courses only go a small way to fulfilling the professional requirements of a software engineer. Zucconi (1995), as cited in Armarego (2004), suggests that software engineers need to be able to work as a member of a multidisciplinary team, to be well organised, engage in lifelong learning and be able to work within the scope of the employer's policies and procedures. Gibson and O'Kelly (2005) add to this list by citing critical thinking, problem analysis, solving complex real world problems and demonstration of versatile and effective communication skills both oral and written. However it is recognised that these attributes need encouragement and support to develop – they do not develop by "osmosis" (Bowden and Marton, 1999).

These generic attributes are very similar to those of traditional engineering students (e.g. civil, electrical etc). This is traced back to the very beginning of software engineering when, in 1968, Peter Naur coined the term "software engineering" at a NATO conference and pointed out that software development should follow an engineering paradigm (*Learning Computing History,* 2004). Ford (1990) defines software engineering as 'that form of engineering that applies the principles of computer science and mathematics to achieve cost-effective solutions to software problems' (as cited in Armarego, 2004). The IEEE describes software engineering as 'the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software' (IEEE, 1999). These basic definitions have been covered by eleven principles as part of the IEEE – CS/ACM SWEEP project which state:

*The professional practice of software engineering encompasses a wide range of issues and activities including problem solving, management, ethical and legal concerns, written and oral communication, working as part of a team and remaining current in a rapidly changing discipline (Delany and Mitchell, 2002).*

Engineering accreditation bodies worldwide now clearly state the need for graduates to be able to function in a team with all the necessary team-work skills this entails (such as conflict resolution and communication) and to deliver solutions which are socially and culturally sensitive and keep abreast with current professional practice. The literature also goes on to suggest that desirable graduate attributes should be expanded to include: working globally in a multicultural environment; working in interdisciplinary, multi-skilled teams; sharing of work tasks on a global and around the clock basis; working with digital communication tools and working in a virtual environment (Thoben and Schwesig, 2002; National Academy of Engineering, 2004).

Many educationalists have tried to deliver courses to software engineering students that replicate the work situation of the profession. Macauley and Mylopoulous (1995) acknowledge that a standard university lecture cannot achieve what industry requires. Students also believe that industry best practice and state of the art technology should be incorporated into the undergraduate curriculum so that they are well prepared for a professional career. They believe that real project work is well regarded by potential employers (Hogan et al., 2005). As these real projects are large and complex, the elements of teamwork and project management are easily integrated. Students often complain, however, that they are never given any advice or guidance on how to work in a team (Hart and Stone, 2002).

## Team work

In PBL, the *team* is the key component for successful student outcomes and individual learning. The team must be committed to a common goal, work interdependently and be mutually accountable. Team members have complimentary skills which support mentoring and peer assistance and they share responsibilities and tasks in line with individual learning goals.

In PBL the basis for learning the required technical concepts is the interaction of team members in delivering the required outcomes of the project. There are many factors which influence the effectiveness of a professional team and not all of these can be addressed within the undergraduate environment (Hogan and Thomas, 2005). Support offered by facilitators or instructors usually centres on communication and time management (Delaney et al., 2003; Armarego, 2004; Hogan and Thomas, 2005). The advantages offered by the USQ curriculum is that the foundation skills of teamwork and communication are covered in the existing problem-solving strand. These co- and pre-requisite courses also incorporate the skills of conflict resolution, leadership, self directed and peer assisted learning and reflection (Brodie, 2007a). Thus the development of any course which uses PBL can build on and extend these skills and it can have an overarching focus on the technical content. Due to exposure in other PBL core courses, students undertaking a software engineering course using PBL should already be proficient in time management, self-directed learning and conducting both virtual and face to face meetings.

Research from the first PBL courses shows that 81% of students agreed that the course increased their ability to work in a team; 73% of students agreed that the ability to learn independently increased and 79% believed their communication skills had increased. These are the averaged results from five years of data (Brodie, 2007a).

### Team selection

The literature varies on methods of team selection. The two most common items are the three core personality types: task-orientated, self orientated, and interaction oriented (Bass and Dunteman, 1963) and the Myers-Briggs profile (Myers, 1998), as cited in Thomas, 1999). Thomas (1999) also investigates the use of Belbin team roles and concludes that these do not seem to predict the success or failure of a team within software engineering projects. Other literature reports that team members are selected so that academically weaker students gain the advantage of working with academically stronger peers (Delany and Mitchell, 2002; Delaney et al., 2003).

At USQ, in existing PBL courses, differing methods are used to select teams. In the foundation course, after several years of random selection of students, the teaching team now uses a "skills audit" to determine prior knowledge and skills in the commencing student cohort. This ensures a mixture of foundation skills within the team, from report writing to knowledge of mathematics and physics (Gibbings and Brodie, 2006). This has significantly improved mentoring outcomes within teams.

The second and final PBL courses still require teams to be multidisciplinary and a random team selection process is employed. The third course is trialling student groups based on Grade Point Average (GPA). These results are still to be analysed but if this method is successful and proven to be academically sound it could form the basis for team selection in the software engineering course.

This background of PBL at USQ provides the students with different aspects of team working and should be capitalised on in the software engineering course.

## Curriculum development

ELE3401 Software Engineering Design Principles (SEDP) is one of the main courses offered to the third year students who study software engineering in the Faculty of Engineering. SEDP introduces general and advanced concepts and techniques in software engineering, with emphasis on the object-oriented design paradigm. Currently the course is supported by printed USQ course material and a textbook, *Object-Oriented Software Engineering using UML, Pattern and Java*, 2nd edition, by Bernd Brugge and Allen H Dutuoit (2003). The course covers about two thirds of the book.

The aim of the SEDP course at USQ is to provide the necessary skills and knowledge for students to carry out the major tasks during the development of large-scale, high quality software systems. It teaches the processes of requirement elicitation and analysis, system and object design, implementation and software validation and verification as would be used in industry. It covers topics and issues such as software life cycle, UML (Unified Modelling Language) modelling, design patterns, system decomposition, interface specification, rational management and project management. The students learn the practical object-oriented techniques and skills for all the phases in the software life cycle. The content of this course is well suited for adaptation to a PBL-based course.

ELE3401 was taught using traditional didactic teaching. This included three hours of lectures and one hour of tutorials per week. External (distance) students were supplied with printed course material. Additional lecture notes

(PowerPoint slides) and tutorial questions were posted for student access via a Learning Management System (LMS). Some discussion topics were also created by the examiner on the LMS online discussion board and students were encouraged to post questions or their opinions to the discussion board. This course has an average enrolment of only eight to ten students and thus was ideal for the implementation of a new teaching method.

Assessment was via two assignments (20% each) and a final examination (60%). The assignments were similar to two small but unrelated projects. The first assignment examined students' knowledge and skills on requirement elicitation, analysis and system design. Students were given a problem statement of a real world software application. They were required to produce a Requirement Analysis Document (RAD) and a System Design Document (SDD). The students worked in pairs for this assignment. The second assignment, which was done independently, examined their knowledge and skills on software testing. The students were given several programs to apply different unit testing techniques.

## Problems with didactic teaching methods

Although the assignments are very close to the 'real world' problem and the first assignment involves minor aspects of team work, they are relatively small and hence not suitable for larger student teams as there are an insufficient number and depth of tasks available. There is also significant content which is not covered by these assignments, for example object design and mapping models to code. Because of these limitations the students often do not experience the whole picture of the design process and do not critically evaluate their submissions. Software engineering in the real world commonly involves participation in large teams, working on very large scale software projects where proper project management and team work are absolutely essential for the success of the project. Therefore the need to simulate a real life problem when teaching software engineering design is evident.

One of the problems of the didactic teaching method is that students do not have sufficient practice during their education in applying software engineering design concepts to a real software engineering problem found in industry. This causes a weakness in skills transfer between academia and industry (Thomas, 2001; O'Kelly et al., 2006).

Another issue, which is particularly serious when dealing with distance and remote students, is that it is very hard for the examiner to assess whether students have made appropriate progress. If the students do not follow the study materials in the suggested course plan and play an active role in their learning, the first indication that relevant learning objectives have not been met is when the first assignment is submitted. However by this point in time it is too late for appropriate remedial actions to be implemented which would help the student to improve. This is evidenced by class grade records which indicate that failure in the first assignment is often indicative of failure in the course.

## Problem development

In order to address these problems PBL has been applied to the software engineering course, simulating a real world software engineering project. A complex problem is presented to the student teams. They then discuss the nature of the problem and the proper ways to solve it, supported by facilitation from tutors and general resources available for the course. Suitable outside resources are also sourced by the students and employed as required. This scenario would be similar to real projects in the software engineering industry and encourages the development of essential skills such as team work, planning, problem solving, communication and critical thinking which are necessary to succeed in the profession (O'Kelly et al., 2006). The application of PBL to this course also encourages students to become independent learners, giving them skills to cope with changing technologies and their application in the real world.

Central to the PBL curriculum development is the problem or problems which focus student learning, team organisation and assessment of both the team and individual. The problem must be complex and open-ended, but scoped sufficiently around the technical learning objectives so that the students learn the relevant technical knowledge. Thus the construction of a suitable problem for the SEDP PBL course is the most difficult part of the curriculum development. One possible solution is to choose a project which comes directly from the software industry. Liaising with the local IT industry provides not only relevant problems or projects but also begins to link students with potential employers and opens up avenues for research for the academics involved. Consultation with industry partners to supply suitable small projects is an ideal solution, though the required consultation to determine exact specifications can be difficult, especially for distance students. The communication between students and the industry partner needs the establishment of protocols and monitoring to ensure both parties are satisfied.

The other approaches that can be used in problem/project construction is to construct the problem/project totally from scratch using published guidelines (Torp and Sage, 2002). However, again this approach has difficulties. The project needs to cover the required content of the course in a thorough manner and this can take significant effort on the part of the course designers. The preferred approach is to use the case study given in the text for the course (a simplified version which would enable the students to be able to complete the whole project in a team environment within a semester to be able to achieve the learning objectives for the course).

## Course structure and schedule

The real world software problem is provided by the local software industry. (Alternatively, an appropriate final year student project can be used. The final year project is usually intended as an individual capstone project in the student's final year of the programme, but it may be suitable for undertaking by a team of third year students). This project is divided into three small phases and each takes around four weeks to complete (see Table 1).

A team leader and deputy team leader are elected for each project. Weekly team progress reports are submitted by the team leaders indicating progress on the project, individual task allocation and progress, comments on project management and notes on any team or resource problems which need to be addressed. This provides a documented record of team progress and individual contributions. Each project is worth 25% of the final mark and three projects total 75% of the final mark. The remaining 25% is covered in a one hour closed book examination.

PBL actively engages students in their own learning and connects it with existing knowledge (Woods, 1994). This knowledge may be directly connected with the problem and the technical content or it may be supporting knowledge and skills, such as leadership and teamwork. PBL can effectively use the prior knowledge and skills of students and then, through course planning and scaffolding, build new knowledge and skills as shown in Figure 2.

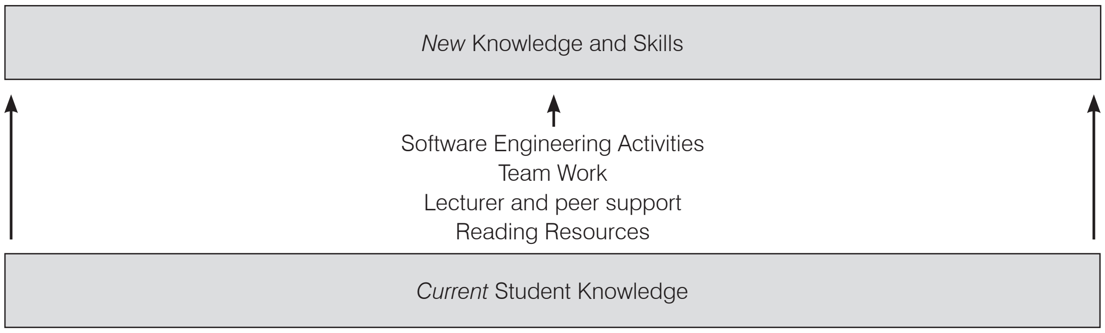
*New* Knowledge and Skills
Software Engineering Activities
Team Work
Lecturer and peer support
Reading Resources
*Current* Student Knowledge

**Figure 2.** Development of student skills and knowledge (Delaney et al. 2003)

**Table 1.** Course project schedule

| Project | Semester Week | Tasks | Products |
|---|---|---|---|
| Phase 1 | 1 - 4 | Requirement elicitation and analysis, system design | Requirements Analysis Document (RAD), System Design Document (SDD) |
| Phase 2 | 5, 8-10 | Object design, implementation | Object Design Document (ODD), coding listing |
| Phase 3 | 11- 14 | Software testing | Testing report |

**Table 2.** Positioning of the software engineering course, supported by other PBL courses

| PBL Course | Position in programme |
|---|---|
| ENG1101 Engineering Problem Solving 1 (core course) | Yr 1 Semester 1 |
| ENG2102 Engineering Problem Solving 2 (core course) | Yr 1 Semester 2 |
| ENG3401 Software Engineering Design Principles (SEDP) | Yr 3 Semester 1 |
| ENG3103 Engineering Problem Solving 1 (core course for BEng students) | Yr 3 Semester 2 |
| ENG4104 Engineering Problem Solving 1 (core course for BEng students) | Yr 4 Semester 2 |

The siting of the course within the programme of study determines much of the starting skills and knowledge of the student cohort. The USQ student cohort at commencing level is very diverse, with students varying in age, entry path, work and study experience (Brodie, 2007a). The existing problem solving strand provides basic skills and scaffolding in self directed learning, teamwork, communication and reflective practice, all of which the software engineering course builds on. It provides an excellent framework in which to situate the 'technical content' demanded of the course with two core PBL courses providing experience of PBL and developing the required supporting skills (see Table 2). Hogan and Thomas (2005) report that after a first year experience of PBL the fundamentals of teamwork have been acquired and subsequent team projects are less directed. This impacts on the type and scope of projects or problems designed for the course.

## Conclusion

The integration of PBL into the software engineering major at USQ uses the existing problem-solving strand as its foundation. This strand of courses provides the students with many of the required foundation skills, including working in multidisciplinary teams, solving complex problems, communication and teamwork and provides an excellent framework in which to situate the 'technical content' demanded of the course.

The use of real world problems has significant benefits, both professionally and educationally. The integration of the 'soft skills' and technical content enhances students' development in both areas. Overseeing, monitoring and mentoring teams can become a much more rewarding task for academics and tutors. However moving from the role of lecturer to facilitator does require new skills. In addition, more time and emphasis need to be placed on the development of problems which incorporate required learning objectives and can be undertaken by both on-campus and distance students working in virtual teams. Thus it is not without considerable planning and resources that a PBL course is developed and delivered. ■

## References

ABET 2000 (2006) Engineering Evaluation Criteria Available from http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/E001%2007-08%20EAC%20Criteria%2011-15-06.pdf#search=%22engineering%20evaluation%20criteria%22 [accessed 16 April 2008].

Aravinthan, T. and Worden, J. (2006) Effective Use of WebCT in a Problem Based Learning Course for a Dual Mode Delivery. EE 2006 *International Conference on Innovation, Good Practice and Research in Engineering Education*, 24-26 July 2006, Liverpool, England.

Armarego, J. (2004) Towards achieving Software Engineering wisdom. *HERSDA 2004*, Miri, Malaysia.

Aspy, D.N., Aspy, C.B. and Quimby, P.M. (1993) What doctors can teach teachers about problem-based learning. *Educational Leadership*, **50** (7), 22-24.

Barrows, H. (2000) *Problem Based Learning Applied to Medical Education*. Springfield: Southern Illinois University Press.

Bass, B. and Dunteman, G. (1963) Behaviour in Groups as a Function of Self. *Journal of Abnormal Social Psychology*, **66** (4), 19-28.

Boud, D. and Feletti, G. (1997) *The Challenge of Problem-Based Learning*, 2nd edition. London: Kogan Page.

Bowden, J. and Marton, F. (1999) *The University of Learning*. London: Kogan Page.

Bridges, E.M. and Hallinger, P. (1992) Problem-based learning in medical and managerial education. *Cognition and School Leadership Conference of the National Center for Educational Leadership and the Ontario Institute for Studies in Education*. Nashville, TN.

Brodeur, D.R., Young, P.W. and Blair, K.B. (2002) Problem Based Learning in Aerospace Engineering Education. *2002 American Society for Engineering Education Annual Conference and Exposition*, Boston, MA.

Brodie, L. (2007a) Problem Based Learning for Distance Education Students of Engineering and Surveying. *ConnectED 2007 International Conference on Design Education*, 9-12 July 2007, Sydney, Australia.

Brodie, L. (2007b) Reflective Writing By Distance Education Students In An Engineering Problem Based Learning Course. *Australasian Journal of Engineering Education*, **13** (1), 31-40.

Brodie, L., Aravinthan, T., Worden, J. and Porter, M. (2006) Re-skilling Staff for Teaching in a Team Context. *EE 2006 International Conference on Innovation, Good Practice and Research in Engineering Education*, 24-26 July, Liverpool, England.

Brodie, L. and Borch, O. (2004) Choosing PBL paradigms: Experience and methods of two universities. *Australasian Association of Engineering Educators Conference*, 26-29 September 2004, Toowoomba, Australia.

Brodie, L.M. (2006) Problem Based Learning In The Online Environment – Successfully Using Student Diversity and e-Education. *Internet Research 7.0: Internet Convergences*, 27-30 September 2006, Brisbane, Australia.

Brodie, L.M. and Gibbings, P.D. (2007) Developing Problem Based Learning Communities in Virtual Space. *ConnectED 2007 International Conference on Design Education*, 9-12 July 2007, Sydney, Australia.

Delaney, J., Mitchell, G. and Delaney, S. (2003) Software Engineering meets Problem-based Learning. *The Engineer's Journal*, **57** (6), 57-59.

Delany, D. and Mitchell, G. (2002) PBL Applied to Software Engineering Group Projects. *International Conference on Information and Communication in Education*,13-16 November 2002, Badajoz, Spain. Available from http://eprints.nuim.ie/archive/00000089/01/JDD_GM_PBL_final.pdf [accessed 16 April 2008].

Engineering Council UK (ECUK) (2003) *Regulating the Profession*. Available from http://www.engc.org.uk/documents/CEng_IEng_Standard.pdf [accessed 16 April 2008].

Felder, R., Woods, D., Stice, J. and Rugarcia, A. (2000) The Future of Engineering Education II. Teaching Methods that Work. *Chemical Engineering Education*, **34** (1), 26-39.

Ford, G. (1990) *SEI Report on Undergraduate Engineering Education, Undergraduate Software Engineering Education* (No. CMU/SEI-90-TR-003). Pittsburgh (PA): Software Engineering Institute/Carnegie Mellon University.

Gibbings, P. and Brodie, L. (2006) Skills audit and competency assessment for engineering problem solving courses. *EE 2006 International Conference on Innovation, Good Practice and Research in Engineering Education*, 24-26 July, Liverpool, England.

Gibbings, P.D. and Brodie, L.M. (in press) Team-Based Learning Communities in Virtual Space. *International Journal of Engineering Education*.

Gibson, J. and O'Kelly, J. (2005) Software engineering as a model of understanding for learning and problem solving. *2005 International Workshop on Computing Education Research*, Seattle, USA. Available from http://portal.acm.org/citation.cfm?id=1089795# [accessed 16 April 2008].

Hart, G. and Stone, T. (2002) Conversations with Students: The outcomes of focus groups with QUT students. *2002 Annual International Conference of the Higher Education Research and Development Society of Australasia (HERDSA)*, 7-10 July 2002, Perth, Australia.

Helbo, J., Knudsen, M., Jensen, L.P., Borch, O. and Rokkjær, O. (2001) Group Organized Project Work in Distance Education. *ITHET 2001 Conference*, 4-6 July 2001, Kumamoto, Japan.

Hogan, J., Smith, G. and Thomas, R. (2005) Tight Spirals and Industry Clients: The Modern SE Education Experience. *Australasian Computing Education Conference*, July 2005, Newcastle, Australia.

Hogan, J. and Thomas, R. (2005) Developing the Software Engineering Team, *Australasian Computing Education Conference*, July 2005, Newcastle, Australia.

IEAUST (1999) *Manual for the Accreditation of Professional Engineering*, Canberra: IEAust.
IEEE (1996) Attributes of the 21st Century Engineer. *Engineering Management Newsletter*, **46** (4), 3-4.
IEEE (1999) IEEE Std. 610-1990 IEEE Standard Glossary of Software Engineering Terminology. *IEEE Standards Software Engineering*, **1**.
Impagliazzo, J. (2004) Learning Computing History 2004. Available from http://www.comphist.org/computing_history/new_page_13.htm [accessed 16 April 2008].
Macauley, L. and Mylopoulos, J. (1995) Requirements Engineering: an educational dilemma. *Automated Software Engineering*, **4** (2), 343-351.
Mayo, P., Donnelly, M.B., Nash, P.P. and Schwartz, R.W. (1993) Student Perceptions of Tutor Effectiveness in problem based surgery clerkship. *Teaching and Learning in Medicine*, **5** (4), 227-233.
Myers, I.B. (1998) *Introduction to Type: a Guide to Understanding your Results on the Myers-Briggs Type Indicator*, 6th edition. Palo Alto, California: Consulting Psychologists Press.
National Academy of Engineering (2004) *The Engineer of 2020: Visions of Engineering in the New Century*. Washington, DC: The National Academies Press.
Newman, M., Ambrose, K., Corner, T., Vernon, L., Quinn, S., Wallis, S. and Tymms, P. (2001) The Project on the Effectiveness of Problem Based Learning (PEPBL): A field trial in Continuing Professional Education. *Third International, Inter-disciplinary Evidence-Based Policies and Indicator Systems Conference*, July 2001, Durham, UK.
O'Kelly, J., Monahan, R., Gibson, J. and Brown, S. (2006) Problem Based Learning: A Software Engineering Curriculum Proposal. *International Conference of Software Engineering*, Maynooth, Ireland
Rhem, J. (1998) Problem-based learning: an introduction. *The National Teaching and Learning Forum*, **8** (1), 1-4.
Rideout, E. and Carpio, B. (2001) The Problem Based Learning Model Of Nursing Education. In: Rideout, E (ed.) *Transforming Nursing Education through Problem Based Learning*. Sudbury: Jones and Bartlett, 21-45.
Sommerville, I. (2007) *Software Engineering*, 8th edition. Harlow, Essex: Addison-Wesley.
Thoben, K. and Schwesig, M. (2002) Meeting Globally Changing Industry Needs In Engineering Education. *ASEE/SEFI/TUB Colloquium*, October 1-4 2002, Berlin, Germany. Available from http://www.asee.org/conferences/international/papers/upload/Global-Education-in-Manufacturing.pdf [accessed 16 April 2008].
Thomas, R. (1999) Group Dynamics and Software Engineering. Object Oriented Systems, Languages and Applications Conference '99, 1-5 November 1999, Denver, USA. Available from http://sky.fit.qut.edu.au/~thomasr/papers/belbin-oopsla.pdf [accessed 16 April 2008].
Thomas, R., Clarke, S. and Adams, M. (2001) Developing Graduate Capabilities through PBL. *Third Asia-Pacific Conference on Problem Based Learning PROBLARC*, 9-12 December 2001, Newcastle, Australia. Available from http://sky.fit.qut.edu.au/~thomasr/papers/developing%20GCs%20thru%20PBL.pdf [accessed 16 April 2008].
Torp, L. and Sage, S. (2002) *Problems as Possibilities*. Victoria: Hawker Brownlow Education.
Vernon, D. and Blake, R. (1993) Does Problem-Based Learning Work? A Meta-Analysis of Evaluative Research. *Academic Medicine*, **68** (7), 550-563.
Wilkerson, L. and Gijselaers, W. (1996) Bringing Problem-Based Learning to Higher Education: Threory and Practice. In: Wilkerson, L. and Gijselaers, W. (eds) *New Directions for Teaching and Learning*. San Francisco, CA: Jossey-Bass, no. 68.
Woods, D. (1994) *Problem based learning: how to gain the most from PBL*. Ontario: Waterdown.
Zucconi, L. (1995) Essential knowledge for the practicing Software Engineer and the responsibilities of university and industry in higher education. *8th SEI Conference on software engineering education*, 29 March – 1 April 1995, New Orleans, USA.

## About the authors

**Lyn Brodie** BEng MEng, (principal author), Senior Lecturer, Electrical, Electronics and Computer Engineering, Faculty of Engineering and Surveying, University of Southern Queensland.
Tel +61746312509   Fax +61746312526   Email: brodie@usq.edu.au

**Hong Zhou** BEng, PhD, Lecturer, Electrical, Electronics and Computer Engineering, Faculty of Engineering and Surveying, University of Southern Queensland.   Email: hzhou@usq.edu.au

**Anthony Gibbons** Undergraduate student, Electrical, Electronics and Computer Engineering, Faculty of Engineering and Surveying, University of Southern Queensland.