

7. TUTORING PROJECT-BASED LEARNING: A CASE STUDY OF A THIRD YEAR SOFTWARE ENGINEERING MODULE AT NUI, MAYNOOTH

**Declan Delaney and George Mitchell,
NUI, Maynooth**

INTRODUCTION

This case study explores teaching and learning on an undergraduate computer science and software engineering degree course called *Real World Software Engineering*. It describes the design, teaching and continuous refinement of the course from an initial concept based on learning-by-doing, to a fully structured course grounded in the Problem-based Learning PBL teaching paradigm (Warner-Weil and McGill, Savin-Baden, 2003; Rhem, 1998; Camp, 1996; Warner-Weil, 1989). Other experiential learning approaches to software engineering education have been explored in Mitchell and Delaney 2004.

In PBL the problem drives the learning process, although experience and research demonstrate that, in general, the assessment process tends to be a greater determinant of what and how students learn (Ramsden, 1992). The *Real World Software Engineering* course described in this study accepts that the problem is the core issue. However, in working through the problem, students encounter a variety of challenges and difficulties they must tackle. These challenges and difficulties motivate student learning. We are specifically utilising the P⁵BL model. The P⁵BL model has five components: Problem- Project- Product- Process- People-Based Learning (Fruchter and Lewis, 2000). The problem is the trigger for the team project work. A software computer engineering product is a deliverable from this project work. The learning and assessment process emphasises process and people skills.

Figure 1: The topics covered in this chapter - The process of reflecting on the design, enactment and assessment of our teaching & continuous feedback to change our philosophy of teaching

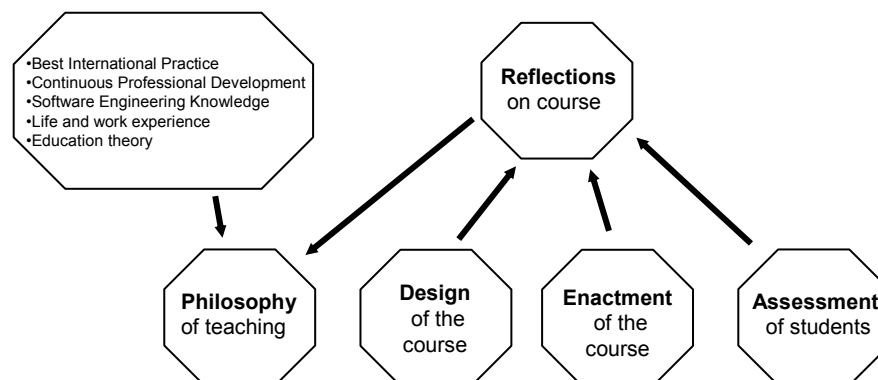
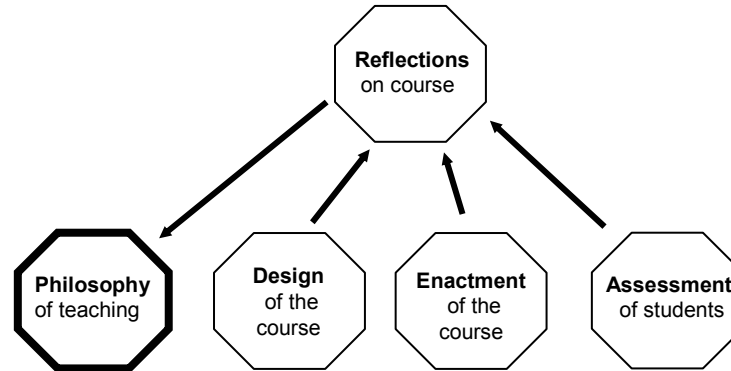


Figure 1 illustrates how our teaching is influenced by our past experiences and continuous feedback from our current teaching experiences. This study outlines the teaching philosophy and objectives of the course as well as their concrete enactment. In addition issues such as the course structure and student assessment and practical problems are explained.

TEACHING OBJECTIVES



Learning is defined as the process whereby knowledge is created through the transformation of experience (Kolb, 1985). This process of transforming experience differs for different people and, as a consequence, people develop preferred learning styles (Pimentel, 1999). According to Kolb's experiential learning theory, there are four elements involved in the learning process: 1 Concrete Experience, 2 Reflective Observation, 3 Abstract Conceptualisation and 4 Active Experimentation (Kolb, 1985). Different disciplines lend themselves more easily to some teaching methodologies rather than others (Schwab, 1964). Awareness of these issues enables us to understand student learning outcomes and the course success more fully.

As teachers we employ both formative and summative assessment techniques when assessing the students and evaluating the overall course. By reflecting on the results of these on-going evaluations, we shape the course so that the student learning outcomes are achieved and we become aware of what we are achieving when we teach. The success of the course is determined by whether the teaching objectives have been met and this success is mainly reflected in the learning outcomes of the students. Have the students gained understanding and command of the subject area? Have they acquired a framework of factual knowledge to allow them construct future knowledge? What have the students accomplished? The answers to these questions all reflect the success or failure of the teaching approach employed.

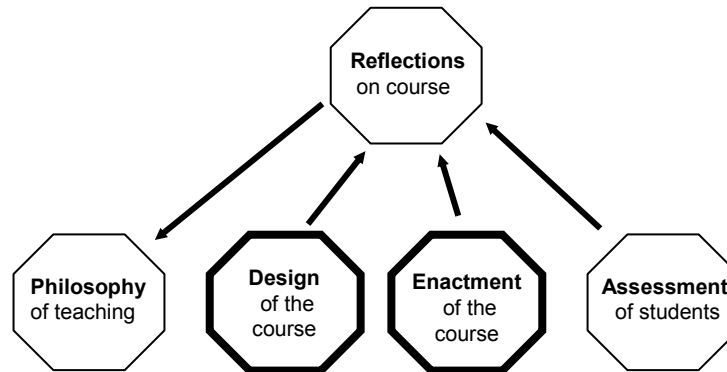
Our specific objectives as teachers of this course are:

- to motivate the students to become interested in the subject of software engineering
- to assist students grasp and master the fundamental software engineering concepts as outlined in the university calendar syllabus
- to apply knowledge and acquire skills by solving a complex software engineering problem
- to raise the students' awareness of international standards and professional bodies
- to foster critical thinking
- to teach students how to learn and acquire life-long learning skills
- to develop problem-solving strategies

Our broader objectives include the all-round education of the students, being aware that how they perform in a learning environment is determined by their physical and spiritual well-being

and that “we cannot reason well in the presence of physical pain, anger, or disappointment or when others do not appear to understand us” (Giussani, 1997).

THE PROJECT-BASED LEARNING COURSE P⁵BL



Software engineering is not so much a series of rules and procedures to be learned by rote as a set of skills that must be acquired and practiced. A highly effective way to achieve this is to get students to work on a long-term, concrete problem which requires them to exercise these skills, exploit knowledge already acquired and acquire new knowledge as needed. This course uses P⁵BL to encourage students to become autonomous learners, to appreciate concrete industrial concerns, to learn to communicate correctly, to become critical thinkers and to become managers of their own learning.

Such an approach is a novelty for most students as there is no end-of-year exam. Students operate in autonomous groups and the teachers are facilitators with the role of guiding and assisting to promote learner independence, while ensuring that the course objectives are being met.

Below we provide details on the overall course structure before examining in detail each of the teaching and assessment methods employed and the practical issues encountered in implementing them. Specifically we discuss:

- course structure
- team building exercises
- group-based development
- presentations and demonstrations
- interviews and consultations
- reflective journals
- formal meetings
- tutorial sessions
- compulsory attendance
- formal documentation
- course website

Course Structure

The course delivery was spread over a 12-week period and consisted of four hours per week contact time with the facilitators and at least four hours per week independent work. The weekly contact time between the facilitators and the students was concentrated into a single

half-day session, with compulsory attendance for all students during the first hour. This gave the students a relatively long period of time to work together and provided them with an opportunity to meet the facilitators under controlled conditions. The learning environment was informal so students could feel free to ask questions and consult the facilitators on any issue.

Team Building Exercises

The initial encounter with the students centred on explaining the teaching philosophy underpinning the course, the course objectives and motivational exercises to promote the advantages of teamwork.


Group-based development: “Real World Software Engineering”

Students were organised into teams of either four or five by the facilitators in order to obtain a weak-strong mix of academic ability in each team. Operating in a team setting, students were asked to solve a complex problem that might typically be found in industry. The problem provided students with an authentic experience of what industrial software development entails. An example is given in Figure 2.

Figure 2: A typical problem specification.

Course: SE315
Real World Software Engineering

The Client's Project Specification


NUI MAYNOOTH
Óliscóil na hÉireann Mhúad

Clients: D. Delaney Rm. 2.112 Tel: 3354
G. Mitchell Rm. 2.109 Tel: 6081

The RESPONSE Project

Our company, WorldWideMail, requires a new web-based email server. This will enhance the services currently being offered by the company and will help us to become the top Remote Email Service Provider, RESPonse. We require that each user have a local user name. Each user should be able to download email or send email from any number of email server accounts supported by our RESPonse web front-end (web browser independent). The front-end should be intuitive and user-friendly. Any downloaded email should be stored and accessible to the user. To attract clients it is essential that we include encryption of all sensitive information such as user passwords. This will make the RESPonse server novel, secure and the best all-round remote email server.

Deliverables

- Documentation completed according to IEEE Standards
- Fully functional software prototype of email server
- A detailed Journal of the implementation process

References

1. A. S. Tanenbaum, Computer Networks, ISBN 0133942481
2. Deitel, Deitel and Nieto, Internet and World Wide Web - How to Program, ISBN 0130161438
3. D. Delaney and S. Brown, Document Templates for Student Projects in Software Engineering, Technical Report NUIM-CS-TR2002-05
4. D. Delaney and S. Brown, Guidelines for Documents Produced by Student Projects in Software Engineering, Technical Report NUIM-CS-TR2002-06
5. <http://hometown.aol.com/nlpjp/index.html>

The facilitators assisted the students in focusing their collective energy into both consolidating and discovering software engineering knowledge and developing software engineering skills. Each team worked on the same problem.

The problems for the projects were sufficiently complex to occupy a group of four/five people for the duration of the course. Initially students were given a very general description of the end product. Most students expressed a sense of the problem being impossible to solve, of being too advanced for third year and of not having enough details to understand fully what they were being asked to do. These reactions were exploited by the facilitators to guide the students in eliciting requirements by teaching them how to ask questions and organise meetings and in designing and testing various software solutions by reminding them of theoretical knowledge they had already acquired. Emphasis was placed above all on the methodology the students employed in approaching the problem, the development of management and communication skills, in addition to the end product of the project.

Presentations and Demonstrations

At the end of the course, each team gave a formal PowerPoint presentation, demonstration and defence of the work they had accomplished. Members of staff were invited to these formal presentations and demonstrations. Following the presentation, the team members were questioned on technical, managerial and team performance issues. A set of standard questions, based on the cognitive domain of Bloom's taxonomy (Davis, 1993), was generated prior to the meetings, so that each group of students was asked the same questions. This assisted in student assessment. Additional questions were asked based on the particular characteristics of each team.

Each presentation and cross examination lasted for one hour. In the first year of the course the presentations lasted for only 20 minutes, but the authors found that the students did not have enough time to adequately describe their work. In addition there was insufficient time to interview the students in depth.

Interviews and Consultations

Individual and team interviews were conducted on a rotating basis during the entire course, with each team meeting with the facilitators at least five times. Consultations were also conducted both inside and outside of the classroom. Facilitators recorded short notes on each student and each team after each consultation and interview. These notes were used as feedback to the learners to assist them in understanding their strengths and weaknesses as well as assisting in final learner assessment.

Time had to be well managed by the facilitators in the classroom. Initially both facilitators attended every interview with every team. However, to ensure an adequate liaison with all teams, we decided that only one of the facilitators would attend each interview and consultation. Teams therefore had separate meetings with both facilitators. The effect of this was that, at times, students received seemingly contradictory advice, as to how best to approach the solution. We believe that this contradictory advice is actually provocative and that it stimulates discussion and exposes the students to different problem solving strategies.

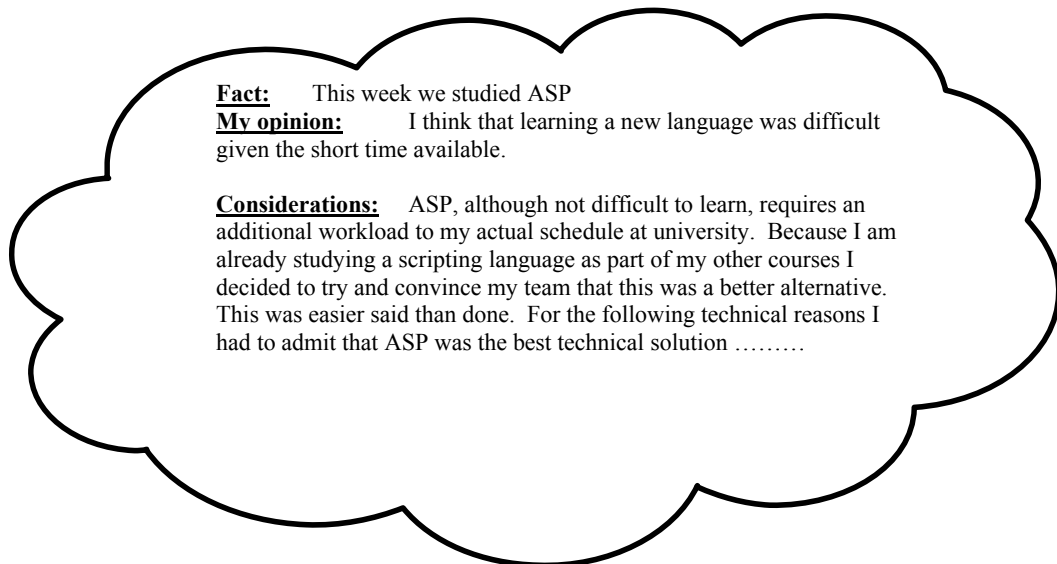
Despite the time and effort involved in conducting and documenting the interviews and consultations, we found them an invaluable source of feedback on student learning and a forum for individual students to raise issues.

Reflective Journals

An individual reflective journal was maintained by each student. This depicted the unique journey of the learner and acted as a forum for the learner to clarify their own thoughts, practice informal writing, develop independent, critical thinking skills and become aware of the learning process occurring. The journal was also valuable as an assessment tool, to distinguish individual learning and achievements from those of the team.

Initially students were reluctant to maintain their journals because they had rarely been exposed to this form of learning before. Students found that writing a reflective journal involved more work than they had anticipated, due more to the reflective process than to the amount of writing requested. In the first year of the course the structure of the journal was left to the discretion of the students themselves. However, the entries were poorly constructed and focused on descriptive commentary rather than on the students' own thoughts and analytical insight. As a result, we introduced a structure for each journal entry to direct the students' thinking process.

Figure 3: An example of the structure of each journal entry



Each entry consisted of a fact, an opinion concerning the fact and reflective considerations. Despite this structure, students still needed much assistance in compiling their journals. As part of the formative assessment procedure, journals were read by the facilitators every three weeks. This meant that constructive feedback and guidance could be offered to the students. The journals of weaker students were also identified and examined more frequently. The journals provided valuable information on the group dynamics and helped us identify potential weaknesses within the groups.

Formal meetings

Learning to participate and conduct a business meeting is an important skill to acquire in many industrial settings and particularly for software engineers who must often manage a team of programmers and liaise between external clients and other software vendors. To facilitate these meetings in a simulated industrial setting, the facilitator assumed the role of client and the student team assumed the role of contractor. Students found that such a structured meeting style was very different to the usual lecturer-student consultations. At the beginning of the course

students had little or no idea of the purpose of a meeting or indeed how to run a meeting. The meetings thus raised the students' awareness of issues such as time management, writing agendas, compiling minutes and chairing discussions. From a lecturer perspective, the meetings provided feedback on the team's development.

Tutorial Sessions

Students, by definition, are not expected to know everything and fundamental gaps in their knowledge thus needs to be bridged. Among the tools chosen to do this was the tutorial session. These sessions were structured by the facilitators, but driven by student questions and the idea was to provide pointers to the students rather than offer definitive explanations. Students learned to ask precise and pertinent questions to drive the tutorial in the direction they desired. Topics for discussion were chosen by consensus among the students themselves and were also determined by the course teaching objectives. The time dedicated to each tutorial varied from 5 to 60 minutes.

The delivery of the tutorials was mainly achieved using the 'chalk and talk' teaching paradigm (Bligh, 2000), although summary slide presentations were made available to the students on the course website. Team teaching was also employed (Buckley, 1999). Both facilitators were present together in the classroom, sharing the floor during teaching sessions, backing each other up and giving students alternative points of view. The students commented on the fact that we didn't always approach a solution in the same way, and this made them aware of different ways to tackle the same problem.

Compulsory Attendance

A compulsory attendance policy can act as a teaching tool in itself. Attendance at the weekly four-hour session was compulsory for the first hour only. After that the students could study independently in a location that best suited their learning needs. During this compulsory hour the facilitators scheduled the tutorial sessions and arranged meetings with particular teams and individuals. As an incentive to attend the lecture we stipulated that if a team member arrived five minutes after the start of the compulsory hour, the following week the entire team would have to give a questions and answers presentation before the entire class. This achieved two objectives: 1 it encouraged time management by individual students and 2 it cultivated a sense of individual responsibility towards other team members. Attendance at lectures during the two years was higher than 95%.

Formal Documentation

In software engineering the ability to formally document the work performed is an essential skill. As end software users, we experience such documentation in the form of user manuals and installation manuals. However, during the software development process such documentation serves to communicate the various phases of the software engineering to other people. For example, the software requirements must be agreed with a client and the subsequent design of the software is based on the documented requirements. Rigorous documentation of testing carried out is essential in all software, but particularly where human safety is an issue. Formal writing, which encompasses international standards, enhances the learner's vision and understanding of the software engineering process and gets the learner to think about how to communicate ideas more effectively.

To assist the students in the arduous task of navigating documentation standards a set of Student Document Templates was developed based on the ISO/IEEE standards (Delaney, 2002; Delaney, 2002).

Course website

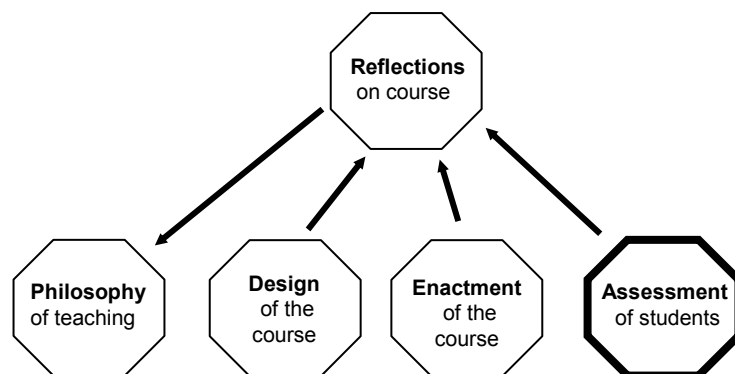
An important instrument for communicating with the students is the course website. This acted as a notice board and a source for recommended reading material such as the university library website.

STUDENT DIVERSITY ISSUES

The diversity of the student body on the course was welcomed because it introduced new ways of looking at old things, it broadened the minds of the students and it made class discussions interesting. The ratio of male to female students was about 6 to 1. Each year there are a number of international students, usually about 5% of the student cohort, and a number of repeat students. The gender issue became evident by observing the roles assumed within the teams. The technical tasks fell almost exclusively to the male component of the team, whereas the task of documenting the work and minutes of meetings fell to the female component. We reduced the tendency towards stereotyped role models by stipulating that all team members must be able to discuss and understand all elements of the project.

The repeat students were also a source of contention. Initially they resented being part of a team with students that began university after them. This led to tension within the teams. In most cases the teams settled down and worked together, but in some cases our direct involvement became necessary. We facilitated team discussion to resolve the difficulties, as well as getting the team members to reflect individually on the issues in their reflective journal.

STUDENT ASSESSMENT ISSUES



Both summative end of course and formative continuous feedback assessment methods were designed into the course. One of the challenging aspects of the assessment procedure was to determine the individual contribution of each student in a group setting. Pure team-based assessment might be a disadvantage to the stronger students and equally be abused by weaker students. The authors therefore implemented a mixture of team and individual assessment techniques, assigning 70% of the marks for group work and 30% for individual work.

The assessment procedure aimed to assess three groups of skills. To assess each skills group, various assessment techniques were employed and staff members other than the mentors participated. The skills groups and assessment techniques can be summarised as follows:

1. *Implementation skills*: ability to implement a software design specification and to produce software documentation based on best practice. The formative assessment occurring during the process of the course examined the delivered product and formal documentation 30% of final mark. The product usually became the main focus of the teams, despite the fact that only 10% of the marks was being awarded for it. A working product is the concrete manifestation of a problem solved, but students needed to be reminded that the key objective of the course is to learn an approach to software development.
2. *Teamwork and leadership skills*: ability to operate in a team environment - how they contribute to the team, how they organise the team and assign roles and responsibilities and how they integrate any industrial experience with their theoretical knowledge. A summative assessment approach student assessment at the conclusion of the course to determine level of skills, knowledge or effectiveness of a plan etc. was employed, incorporating a group presentation and individual interview 40% of final mark. The final presentation was a highly effective assessment technique, despite the time and effort involved.
3. *Analytical thinking, problem solving and interpersonal skills*: ability to think through a problem, analyse a situation, deal with pressure and communicate with the clients. In this case a formative assessment approach based on an individual reflective journal, and a summative technique based on feedback forms peer and self evaluation forms 30% of final mark. The informal journal is excellent for determining individual learning and development. It is important to look for evidence of reflective thought and not descriptive accounts in a commentary mode.

CONCLUDING REMARKS

In western society today, the biggest problem facing scholars is not the lack of information, but sourcing relevant information and applying it in a meaningful manner. Our approach scaffolds student learning in such a way that they learn how to access and apply information in the context of a real-world problem.

Our classroom offers an interactive learning environment that fosters student learning in collaboration with each other and with the facilitators. The course brings software engineering alive and in so doing motivates students to become active and self-directed learners. Our understanding of how the problems contained in P⁵BL impact student learning have been published elsewhere (Delaney, 2005).

Reflecting on what teaching and learning means for us and enquiring into what works and what needs to be reviewed has been a key element to the success of this P⁵BL course and has subsequently changed our attitude to teaching and learning on other courses we teach. Our philosophy of teaching results from an ongoing process of refinement based on constant reflection on our experiences, listening to students and peers, and investigating the theory

underlying education. We trust that our experiences outlined here will be of some practical use for others who endeavour to implement a small project-based course.

ACKNOWLEDGEMENTS

Dr. Aidan Mulkeen, Education Department, NUI Maynooth; Mr. Seán Delaney, Coláiste Mhuire Marino, Dublin; Mr. Stephen Brown, Department of Computer Science, NUI Maynooth.

REFERENCES:

- Bligh, D. A. 2000 *What's the Use of Lectures?* San Francisco: Jossey-Bass.
- Buckley, F. J. 1999 *Team Teaching: What, Why, and How?* London: Sage.
- Camp, G. 1996. *Problem-based Learning: A Paradigm Shift or a Passing Fad?* Medical Education Online 1, 2, 1-6.
- Davis, B. G. 1993 *Tools for Teaching.* San Francisco: Jossey-Bass.
- Delaney, D. and Brown S. 2002a Document Templates for Student Projects in Software Engineering NUIM-CS-TR2002-05. Maynooth: National University of Ireland, Maynooth.
- Delaney, D. and Brown, S. 2002b. Guidelines for Documents produced by Students in Software Engineering NUIM-CS-TR2002-06.
- Delaney, J. D. and Mitchell, G. 2005 "More than Just Problem Solving: Unpacking Problem Based Learning PBL." *The Engineers Journal*, 59, 3, 166-169.
- Delaney, J.D and Mitchell, G.G. 2002 *PBL Applied to Software Engineering Group Projects.* International Conference on Information and Communication in Education. Badajoz, Government of Extremadura, Spain: 1093-1098.
- Fruchter, R. and Lewis, S. 2000 *Mentoring and Reverse Mentoring in P⁵BL.* Paper presented at the 2nd Asia-Pacific Conference on PBL. Singapore.
- Giussani, L. 1997 *The Religious Sense.* Montreal: McGill-Queen's University Press.
- Kolb, D. A. 1985 *Experiential Learning: Experience as the Source of Learning and Development.* New Jersey: Prentice-Hall.
- Mitchell, G. G. and Delaney, J. D. 2004 "An Assessment Strategy to Determine Learning Outcomes in a Software Engineering Problem-based Learning Course." *The International Journal of Engineering Education* 20, 3, 494-502.
- Pimentel, J. R. 1999 "Design of Net-learning Systems Based on Experiential Learning". *Asynchronous Learning Networks*, 3, 2, 64-90.
- Ramsden, P 1992 *Learning to Teach in Higher Education.* London: Routledge.
- Rhem, J. 199. "Problem-based Learning: An Introduction." *The National Teaching and Learning Forum*, 8 1.

Savin-Baden, M. 2003 *Facilitating Problem-based Learning: Illuminating Perspectives*. Berkshire: SRHE/Open University Press.

Schwab, J. 1964 *Structure of the Disciplines: Meanings and Significances*, in G. W. F. A. L. Pugno ed. *The Structure of Knowledge and the Curriculum*. Chicago: Rand McNally.

Warner-Weil, S. and McGill, I. 1989 *Making Sense of Experiential Learning: Diversity in theory and practice*. Milton Keynes: Open University Press.