2015-4

An Evaluation of the Effectiveness of Using a Hybrid PBL Approach in the Teaching of the Java Programming Language to First Year Third Level

James Doody
Institute of Technology Blanchardstown, james.doody@ittdublin.ie

Funder: Tu4D

An evaluation of the effectiveness of using a hybrid PBL approach in the teaching of the Java programming language to first year third level students

James Doody
ITT Dublin
james.doody@ittdublin.ie

Abstract

First year students on third level Computing courses find Software Development difficult: learner outcomes are poor, with high failure rates and low learner retention. A number of research studies have shown that novice programmers have low intrinsic motivation and low programming self-efficacy. One of the other possible explanations for the difficulties many learners have with Software Development is that it may be a Threshold Concept in Computing. The literature suggests that Problem-Based Learning (PBL) can improve the teaching of difficult concepts, and it has been promoted by professional and funding bodies as a teaching strategy that can improve learner outcomes and bring about positive changes in learner behaviour. The main aim of this research study was to establish the impact on learner behaviour of a Hybrid PBL approach used in the teaching of an introductory Software Development module at an Irish third level institution. Learners on the Software Development module are characterised by low prior attainment in State college entry examinations, and the majority are from low income socio-economic backgrounds. Learner behaviours were investigated over four cohorts of learners using a large range of data sources. A randomised controlled experimental design was used to measure changes in attainment, programming self-efficacy, motivation, approaches to study and preferences for types of teaching. Questionnaires, data mining of learner activity and attendance logs were used to provide additional information about learner behaviour, and further analysis was undertaken using qualitative techniques such as classroom observations and interviews. Both qualitative and quantitative measures were used to confirm, cross-validate and corroborate findings. The study made significant discoveries about the strengths and limitations of the Problem-Based Learning approach in the teaching of Software Development to low attainment learners. The implications for instructional practice and for educational theory and research are discussed and a number of recommendations are made.

Keywords: Problem-Based Learning, Software Development, Computer Programming, Curriculum, Programming Self-Efficacy, Motivation, Approaches to Studying, Teaching, Learning.

1. Introduction and Rationale

The production of defect-free quality software is essential for the correct operation of many critical systems. The demand for software is growing and it has become ubiquitous. However, there are many problems with the production of software, in particular it is often poorly written and faulty. More time is spent fixing errors in existing software than writing new code. The economic cost of software failure is counted in billions: in the U.S. alone, software bugs cost the economy an estimated $59.5 billion annually (Newman, 2002). There are many causes of software failure and key among them are the deficits in the education and training its creators received. Software is not a mass-produced product: it is handmade, crafted by individuals. Most of these individuals are educated as Software Developers in universities and other higher education institutes and they require mastery of a diverse range of skills to become competent programmers (Lohr, 2001). It is well accepted within the computer science community that first year students find Software Development difficult (Dijkstra, 1989; Jackson, 2003; Jenkins, 2002). Failure rates are high and learner retention is low (Bennedsen &

Caspersen, 2007). Many learners have low intrinsic motivation (Mamone, 1992). Many students show high reproduction orientation in their approaches to studying (Jenkins, 2001). Novices' programming self-efficacy levels are low (Wiedenbeck, LaBelle, & Kain, 2004); and improvements need to be made in the way that Software Development is taught (Fincher, 1999; Fincher et al., 2005; Jenkins, 2002). Recent educational research may help provide some solutions to these problems. A number of research papers have identified that Software Development (Java programming) is a Threshold Concept in Computing (Boustedt et al., 2007; Eckerdal et al., 2006). The literature suggests that Problem-Based Learning can improve the teaching of difficult concepts (Ayres, 2002; Hmelo-Silver, 2004; O'Kelly, 2005) and bring about improvements in learner behaviour (Dolmans & Schmidt, 2006; Richardson, 2005; Schmidt, Loyens, van Gog, & Paas, 2007). Newman (2004, p. 1) states that Problem-Based Learning (PBL) "represents a major development and change in educational practice that has a broad impact across subjects and disciplines worldwide". PBL is promoted by professional and funding bodies as an appropriate strategy for education and increasingly as a method of choice.

While there have been a number of informative Irish PBL case studies (e.g. Barrett, Mac Labhrainn, & Fallon, 2005), none has focused on low attainment learners on Software Development Programmes. According to Barrows & Tamblyn (1980, p. 1):

> "Problem-based learning is the learning that results from the process of working toward the understanding or resolution of a problem. The problem is encountered first in the learning process."

PBL is a teaching method that can be used in many formats, such as small-group tutorials, problem-based lectures, large-group case method discussion, and problem-based laboratories (D. M. Kaufman, 1995). However, it is used most commonly in small groups with a facilitator. The essence of the PBL method involves three steps: confronting the problem; engaging in independent study; and returning to the problem (Wilkerson & Feletti, 1989). While some aspects of PBL are considered highly effective, the effectiveness of other aspects is disputed (Albanese & Mitchell, 1993; Berkson, 1993; R. Butler, Inman, & Lobb, 2005; Mike Newman, 2004; Norman & Schmidt, 1992; Vernon & Blake, 1993).

One area of debate is the effect of PBL on learners' acquisition of knowledge and skill (the application of knowledge). Albanese and Mitchell (1993) found shortfalls in students' knowledge following PBL courses compared with students enrolled on traditional courses. This shortfall in knowledge is supported by the findings of other studies (Baca, Mennin, Kaufman, & Moore-West, 1990; Eisenstaedt, Barry, & Glanz, 1990). An analysis of the impact of Hybrib PBL on learner attainment was also undertaken and the findings reported elsewhere (Doody, 2009).

PBL has been implemented in environments varying in scope from one single course (Lewis & Tamblyn, 1987) up to an entire curriculum (A. Kaufman et al., 1989). As Dochy et al state "while the impact of PBL as a curriculum is certainly going to be more profound, a single course can offer a more controlled environment to examine the specific effects of PBL". This view is shared by other researchers (Albanese & Mitchell, 1993; Schmidt, 1990). This is the approach we have taken in this study.

2. The Introduction of Hybrid PBL.

In the mid-2000s, there was a general decline in the number of second level students choosing Computing at third level. This led to the entry into first year Computing of low attainment learners. This in turn led to a problem of poor student retention in first year, with the Software Development module having particularly high failure rates. It is well accepted within the computer science community that first year students find Software Development difficult (Jackson, 2003). One of the major stumbling blocks is the abstraction of the problem to be solved from the exercise description (McCracken et al., 2001). It was considered that if a new way of teaching the Software Development module were introduced, the high failure rates in that subject could possibly be redressed and first year retention rates ultimately improved. The Computer Science Department at another Irish College (O'Kelly, 2005) had already introduced a PBL model to teach first year Software Development and it was decided to apply the same model, including lecturers being provided with PBL training. Training helps to initiate and develop the PBL programme and to assist staff in adjusting to the role of facilitator/mentor/coach (Donald R. Woods, 1996).

2.1 Implementation of the Hybrid PBL module

Ellis and Dick (2000) argue that group size has a number of effects, including the degree of participation possible and the strength of bonds between members. Groups of 7-8 students were decided upon. Gender balance was difficult to achieve with approximately 90% of the class being male each year. Each team developed its own set of ground rules for behaviour and goal achievement, and these rules were reviewed regularly by the team. It was the responsibility of each team to keep its journal updated. Each team worked together for the entire semester.

The problems used to teach the PBL module were developed by O'Kelly (2005) and are based around specific Software Development learning outcomes. The problems created fall into three broad categories: firstly, extendable conceptual problems, that is, problems that ensure the students focus on core concepts of computer programming in order to solve a problem. These problems involve no programming but require that the students understand programming-related concepts. The problems also allow for increased levels of difficulty to be added to the problem once a solution is found to ensure that the problem sustains the students' interest. The second category of problems used is non-extendable conceptual problems which help a student to understand programming-related concepts without performing any programming. This type of problem has just one solution and is not extendable. The third category of problems, programming problems, are typical computer programming problems that the group tries to solve collectively. This type of problem aids the weaker student as he/she gets to see how a stronger student solves a programming problem (O'Kelly et al., 2004).

While the amount of lecture time provided for Software Development remained unchanged under the PBL model, the structure of the lectures was changed. The PBL approach used copied that used by O’Kelly (2005) and was informed by the work of Deek et al (1993), Woods (1996) and Waite et al (2003). Under this hybrid PBL model, a problem was presented at the beginning of class: the students were paired and asked to generate possible ideas to solve the problem. Each pair of students was then grouped with another pair and this bigger group was asked to develop an algorithmic solution based on their combined ideas. The lecturer facilitated the group process during this period. The lecturer then collaborated with the students to solve the problem algorithmically with ideas generated from different groups of students. Once a solution to the problem was drafted, the lecturer stepped through the solution with the students, any difficulties were identified and rectified by the class and the step-through process began again until such time as a viable solution was reached. At this point the translation of the algorithm to code occurred. During this process any programming concepts that students did not understand were flagged and covered in tutorials. The methods used to assess the students summatively remained unchanged under the PBL model (two in-laboratory based practical assignments and a paper-based closed book end-term exam).

3. Research Methodology

From a detailed literature review a number of research questions emerged (Doody, 2009). These research questions were examined in detail in the context of the implementation of a hybrid PBL Java programming module for novices at an Irish higher education establishment. Attainment related research questions are discussed elsewhere (Doody, 2009).
This paper examines:

1. What are the effects of using a PBL model on learner self-regulation?
2. What are the effects of using a PBL model on learners’ programming self-efficacy?
3. What are the effects of using a PBL model on students’ approaches to learning and on general learner engagement?
4. What are the effects of using a PBL model on learner preferences for different types of course and teaching?

3.1. Participants

Participants in the study were drawn from four cohorts of first-year students who enrolled for the academic years 2005/2006, 2006/2007, 2007/2008, and 2008/2009. In all, 398 first year students took part in the study. Repeat students taking the module for a second time were excluded from the study, therefore, each year the cohort contained a different set of participants from the previous year. Demographic details show a learner population profile with a male:female ratio of around 10:1, with all students speaking English as their first language, almost all of Irish nationality, all except one learner between 18 and 23 years of age, and the majority living in areas of Dublin suffering from socioeconomic disadvantage. Ten lecturing staff and four tutors also took part in the study.

3.2. Methods of Analysis

A mixed method design including both qualitative and quantitative measures was used in this study. A concurrent triangulation strategy was employed to add validity to the research findings (Creswell, 2003, p. 215). A quantitative, controlled, experimental research design was used to empirically test each of the research questions. In addition, a qualitative approach based on grounded theory was used to further explore and scrutinize each research question.

Learner participants were randomly split into a PBL treatment group (Group A) and a non-PBL treatment control group (Group B). This was done for each of the four cohorts. Each hypothesis was tested quantitatively over a number of cohorts using the instruments (described in section 3.3 below) which were given out before and after the teaching, and effect sizes for each hypothesis were calculated. In addition, information on learners' attendance was analysed statistically. Qualitative information on learners' backgrounds and PBL experiences was collected using questionnaires. Furthermore, interviews were carried out with learners and staff involved in the PBL group and detailed field notes were taken of observations of learner in-class behaviour.

3.2.1. Controls

The same staff member acted as overall coordinator for the module for the duration of the study. This allowed for the control of teacher effects. The same methods of summative assessment were used for all four cohorts of learners. Within each cohort, identical marking schemes and assessments were used for both groups. The physical learning environment of classrooms and computer laboratories, and the time allocation and combination of lectures, tutorials and laboratories was the same for both groups. In cases where statistical tests assumed a normal distribution of data, the Kolmogorov-Smirnov normality test was carried out to ensure normality.

3.3 Instruments and Measures

A review of the literature identified a number of established instruments that could be used to help test the different hypotheses (Doody, 2009).

3.3.1. Learner Self-Regulation (Research Question 1).

Participants' Learning Self-Regulation (Autonomous or Controlled Regulation) was measured over two cohorts of learners using a statistical analysis of learner responses on the Learning Self-Regulation Questionnaire (SRQ-L) (Williams & Deci, 2007) which was given out to all participants in both groups at the start and end of semesters 1 and 2. The Self-Regulation Questionnaires used are well validated, (Black & Deci, 2000; Ryan & Connell, 1989; Williams & Deci, 2007).

An indicator of the success of the Hybrid PBL model is whether Group A's Computer intrinsic motivation increased at a greater rate than Group B's. A number of statistical tests were carried out on the results to assess changes in learners' intrinsic motivation due to attending the PBL module:

- Group A's Learning Self-Regulation results at the start and finish of semester 1 were compared against Group B's results;
- Any change in Group A's Learning Self-Regulation during semester 1 was compared against Group B's results.

From these comparisons it was possible to test the following hypothesis:
1. Learners who complete the PBL course will have a higher degree of intrinsic motivation than those in the control group.

3.3.2. Programming Self-Efficacy (Research Question 2).

Participants' Programming Self-Efficacy was measured over two cohorts of learners using a statistical analysis of learner responses on the Computer Programming Self-Efficacy instrument (PSE) (Ramalingam & Wiedenbeck, 1998), which was given out to all participants in both groups at the start and end of semesters 1 and 2. The PSE Scale was developed for use with object-oriented programming languages, and has been used in a number of studies in higher education on learners of the Java programming language (Askar & Davenport, 2009; Bergin & Reilly, 2005; Cantwell-Wilson & Shrock, 2001; Ramalingam & Wiedenbeck, 1998).

An indicator of the success of the Hybrid PBL model is whether Group A's Computer Programming Self-Efficacy increased at a greater rate than Group B's. A number of statistical tests were carried out on the results to assess changes in learners' Self-Efficacy due to attending the PBL module:

- Group A's Computer Programming Self-Efficacy results at the start and finish of semester 1 were compared against Group B's results.
- Any changes in Group A's Computer Programming Self-Efficacy during semester 1 was compared against Group B's results.

From these comparisons it was possible to test the following hypothesis:
2. Learners in the PBL group will show a higher degree of programming self-efficacy than those in the control group.

3.3.3. Students' Approaches to Learning and Learner Preferences (Hypotheses 3 & 4)

Students' approaches to studying and learner preferences were measured over two cohorts of learners, using a statistical analysis of learner responses on parts B and C of the Approaches and Study Skills Inventory for Students (ASSIST), which was given out to all participants in both groups at the start and end of semesters 1 and 2. ASSIST was developed by Entwistle (1997) and has been widely used, is well validated, and has had its reliability well tested (Entwistle, Tait, & McCune, 2000; Long, 2003; Tait & Entwistle, 1996). An indicator of the success of the Hybrid PBL model is whether learners in Group A show higher scores on meaning orientation and lower scores on reproduction orientation and a greater preference for teaching that supports deep learning than learners in Group B. A number of statistical tests were carried out on the results to measure and assess changes in learners' learning orientation and preferences due to attending the PBL module:

- Group A's approaches to study and their preferences scores at the start and finish of semester 1 were compared against Group B's results;
- Any change in Group A's approaches to study and preference scores during semester 1 was compared against Group B's results.

From these comparisons it was possible to test the following Hypotheses:
3. Learners in the PBL group will show higher scores on meaning orientation and lower scores on reproduction orientation than those in the control group.
4. Learners in the PBL group will show a greater preference for courses and teaching that support deep learning (as opposed to surface learning) than those in the control group.

4. Results

4.1. Motivation

Some evidence was found that the hybrid PBL model brought about a slight improvement in learners' relative autonomy with an overall effect size of (ES = 0.23). Nonetheless, given that the results are not statistically significant, it cannot be said that learners who complete the PBL course will have a higher degree of intrinsic motivation than those in the control group. This was an unexpected result in view of the research that suggests that the PBL teaching method promotes perceived autonomy and self-determination (S. Butler, 1999; De Volder, Schmidt, Moust, & De Grave, 1986; van Grinsven & Tillema, 2006), which in turn can have a positive effect on students' motivation (Deci & Ryan, 1985; Hidi & Harackiewicz, 2000). However, one major difference between those studies and this study is that the participants in the former were high attainment learners. In addition, given that research has shown low levels of intrinsic motivation and high levels of extrinsic motivation to be attributes of learners on programming courses (Mamone, 1992), research is needed to examine if learners on certain Computing courses are less intrinsically motivated than learners on high status courses like medicine.

4.2. Software Development Self-Efficacy

Evidence was found that the hybrid PBL model brought about a significant improvement in learners' programming self-efficacy with an overall effect size of (ES = 1.70). Therefore it can be said that learners who complete the PBL course will have a higher degree of programming self-efficacy than those in the control group. This result was expected given the research that shows a link between programming self-efficacy and PBL (Bergin & Reilly, 2005; Dunlap, 2005), and programming self-efficacy and improved performance in skills (Wiedenbeck et al., 2004). To explain this finding, it might be the case that the specific instructional strategies used in PBL, namely the use of authentic problems of practice, collaboration and reflection, increase student engagement and are therefore the catalysts for students' improved self-efficacy (Hendry, Frommer, & Walker, 1999). The effect size in this study was larger than that reported by Bergin and Reilly (2005) in a study at an Irish university on the role of comfort-level (including programming self-efficacy) on a first-year object-oriented Java programming module taught using a Problem-Based Learning approach. This divergence in findings might be partially explained by the difference in prior attainment of the participants. Given the low prior attainment of learners in the study, it is possible that they had greater scope for improvement in programming self-efficacy.

4.3. Approaches to Studying

When compared against the non-PBL group there was evidence that the hybrid PBL model led to an improvement in learners' meaning orientation, with an overall effect size of (ES = 0.35) on deep approaches to learning, and a reduction in reproduction orientation with an effect size of (-0.75) on surface apathetic approach. A small negative effect was also seen on the strategic approach, with an effect size of (ES = -0.41). From these findings it can be said that learners in the PBL group will show higher scores on meaning orientation and lower scores on reproduction orientation than those in the control group. This result was expected and is in line with the results of studies of paramedical and medical students students (Newble & Clarke, 1986; Sadlo, 1997).

4.4. Preferences for Different Types of Teaching

Evidence was also found that the hybrid PBL model led to an increase in learners' preference for Supporting Understanding approaches to teaching with an overall effect size of (ES = 0.36) and a reduction in learners' preference for Transforming Information approaches to teaching with an effect size of (-0.63). These results suggest that learners in the PBL group will show a greater preference for courses and teaching that support deep learning (as opposed to surface learning) than those in the control group. These findings are in line with results from other studies that show evidence that PBL enhances students' approaches to learning and improves their perception of the quality of their course (Sadlo, 1997; Sadlo & Richardson, 2003).

4.5. Discussion of Other Findings

Feedback from interviews with learners suggests that the PBL model used may provide a good transition for students to a third-level environment by helping them get to know the other students in their class. It also facilitates students in developing peer group support networks that help to remove the feelings of isolation commonly experienced by first-year students. However, some students said during interview that they did not like the course structure and some students said that they did not feel that they had actively participated in the PBL sessions. A group size of 7-8 may be too large and allow some members to avoid working on the problems. Tutors need to be aware of these difficulties and provide independent work in the laboratories and closely monitor the division of work within PBL groups.

Observations of the PBL labs showed that they were in general active learning environments. However, questionnaire responses showed that students did not spend much time outside of class revising software topics or problems. This suggests that students do not reflect on their learning activities outside of class time. The observations of the PBL labs as active learning environments support the finding that learners in the PBL group will show higher scores on meaning orientation and lower scores on reproduction orientation than those in the control group, and that they will also show a greater preference for courses and teaching that support deep learning (as opposed to surface learning) than those in the non-PBL group. However, the observation that the PBL learners did little further work outside class time shows that these effects are limited.

The non-PBL groups showed little inter group tension, with students chatting and laughing about their social activities. The PBL groups displayed some limited intra group tension and arguments, and a number of students felt that the group climate did not facilitate the learning process. This point was raised particularly by the female students. The issues they raised were that some male members of the group did not contribute to the problem-solving and that the females did not like engaging in arguments about group activities. Other studies have also identified issues of an unfair distribution of work in PBL groups (Kinnunen & Malmi, 2005; Donald. R. Woods, Hall, Eyles, Hrymak, & Duncan-Hewitt, 1996), and strategies need to be identified to address this problem.

PBL groups that worked efficiently had focused discussions about programming problems: their conversations did not lapse into irrelevant topics. Inefficient PBL groups also had members who were very dominant due to their previous knowledge or their personality. Tutors need to be aware of this problem and can help other students to cope with dominating students in constructive ways. Other studies have also identified the problem of dominant group members and they provide guidance for tutors in addressing this problem (Benbow & McMahon, 2001; Donald R. Woods, 1996). PBL group tensions have also been noted by Kinnunen & Malmi (2005) who conducted a study of PBL in an introductory programming course in Finland.

Another issue highlighted by the study was staff burnout. When PBL was first introduced in the Computing Department, staff were particularly enthusiastic and devoted a great deal of effort to its organisation and delivery. In the following years the enthusiasm lessened, mainly due to the high workload involved in supporting the PBL classes (Doody, 2009). Marsh (1987) reports that this is a common occurrence when PBL is introduced. It is also interesting to note that dropouts were spread evenly between both the PBL and non-PBL group, a finding which is contrary to what Newman (2004, p. 151) observed in his meta-analysis of PBL, where dropout rates were much higher in the PBL groups.

5. Discussion, Limitations and Conclusions

> “[T]he answer to the question ‘Does PBL work?’ is: it depends.”
> (Richardson, 2005, p. 51).

5.1 Limitations of the Study

There are some limitations of this study that must be taken into account before reaching any generalisations. First, the learners in this study were mainly low attainment learners and the findings may not be more generally applicable to contexts involving high attainment learners. Second, the groups are not totally statistically independent, as Computing students mix freely between groups and with engineering students outside of class time. Third, most of the learners in the study were grant-aided in that they are paid for attending classes. This may skew attendance and retention rates and lessen the general applicability of the findings. Fourth, some of the findings in this study are based on learner responses on self-report questionnaires. However, a number of steps were taken to ensure validity. Fifth, learner participants in this study were very homogeneous: there was a small number of female and ethnic minority participants, and the needs of students with disabilities and special educational needs were not focused upon. Finally, it should be noted that the sample frame used in this study, constituted an opportunity sample, and that the finding cannot therefore be safely generalized to higher education as a whole. A multi-national, multi-institutional study would provide more generalisable findings and overcome some of the possible shortcomings of using an opportunity sample.

5.2. Conclusions

Although it cannot be said that learners who complete the PBL course will have a higher degree of intrinsic motivation than those in the control group, the comparisons between groups provide support for the hypotheses that first year Software Development students taught using a PBL approach will: have a higher degree of programming self-efficacy than those in the control group; show higher scores on meaning orientation and lower scores on reproduction orientation than those in the control group; show a greater preference for courses and teaching that support deep learning (as opposed to surface learning) than those in the control group; and perform better in continuous assessment that test skills but not in final exams that test knowledge.

The improvement in skills is perhaps because in the non-PBL group the learning effort was mainly focused on programming strategies focused on code syntax and a trial and error attempt to develop a correct programme schema, while in the PBL group the learning effort was mainly focused on developing programming strategies based on a correct programme schema, and not on code syntax.

The study provides evidence that the PBL model assists students in problem abstraction, problem definition and problem refinement. Interviews with staff suggest that the non-PBL group working on the same set of problems remained stuck dealing with syntax issues, rather than mastering the concepts of abstraction and object orientation. Thus it is likely that the students taught using the PBL method will develop greater mastery of the concepts of object orientation and abstraction. This suggests that the PBL method is better at helping students master Threshold Concepts in Computing, which in turn suggests that the use of PBL to teach novice learners may help to improve student retention.

References

Albanese, M. A., & Mitchell, S. (1993). Problem-based learning: a review of literature on its outcomes and implementation issues. *Academic Medicine, 68*(8), 52-81.

Askar, P., & Davenport, D. (2009). An Investigation of Factors Related To Self-Efficacy For Java Programming Among Engineering Students. *The Turkish Online Journal of Educational Technology, 8*(1).

Ayres, F. (2002). Problem-based learning: the benefits to students and organisations. *Training Journal*, 20-22.

Baca, E., Mennin, S. P., Kaufman, A., & Moore-West, M. (1990). Comparison between a problem-based, community-oriented track and a traditional track within one medical school. In Z. H. Nooman, H. G. Schmidt, & E. S. Ezzat (Eds.), *Innovation in medical education: An evaluation of its present status* (pp. 9–26). New York: Springer.

Barrett, T., Mac Labhrainn, I., & Fallon, H. (Eds.). (2005). *Handbook of Enquiry and Problem-based Learning: Irish Case studies and International Perspectives* (1st ed.). Galway: CELT NUI Galway.

Barrows, H., & Tamblyn, R. (1980). *Problem-based learning: an approach to medical education.* New York: Springer Publishing Company.

Benbow, E. W., & McMahon, R. F. T. (2001). Mature students? In P. Schwartz, S. Mennin, & G. Webb (Eds.), *Problem-based Learning. Case Studies, Experience and Practice* (pp. 119-125). London: Kogan Page.

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin, 39*(2), 32-36. doi: http://doi.acm.org/10.1145/1272848.1272879

Bergin, S., & Reilly, R. (2005). *The influence of motivation and comfort-level on learning to program.* Paper presented at the 17th Workshop of the Psychology of Programming Interest Group, Sussex University.

Berkson, L. (1993). Problem-based learning: have the expectations been met? *Academic Medicine, 68*(10 Supplement), S79-88.

Black, A. E., & Deci, E. L. (2000). The effects of instructors' autonomy support and students' autonomous motivation on learning organic chemistry: A self-determination theory perspective. *Science Education, 84*(6), 740-756.

Boustedt, J., Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & Zander, C. (2007). Threshold concepts in computer science: do they exist and are they useful? *ACM SIGCSE Bulletin, 39*(1), 504-508. doi: http://doi.acm.org/10.1145/1227504.1227482

Butler, R., Inman, D., & Lobb, D. (2005). Problem-based learning and the medical school: another case of the emperor's new clothes? *Advances in Physiology Education*(29), 194-196.

Butler, S. (1999). Catalysing student autonomy through action research in a problem centred learning environment. *Research in Science Education, 29*(1).

Cantwell-Wilson, B., & Shrock, S. (2001). *Contributing to success in an introductory computer science course: a study of twelve factors.* Paper presented at the 32nd SIGCSE technical symposium on Computer Science Education Charlotte, North Carolina, United States.

Creswell, J. W. (2003). *Research Design: Qualitative, Quantitative, and Mixed Method Approaches* (2nd ed.). London: Sage Publications Inc.

De Volder, M. L., Schmidt, H. G., Moust, J. H. C., & De Grave, W. S. (1986). Problem-based-learning and intrinsic motivation. In J. H. C. van der Berchen, T. C. M. Bergen, & E. E. I. de Bruyn (Eds.), *Achievement and task motivation* (pp. 25-32). The Netherlands: Swets.

Deci, E. L., & Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. New York: Plenum Press.

Deek, F. P., & Kimmel, H. (1993). *Changing the Students' Role: From Passive Listeners to Active Participants.* Paper presented at the 23rd Frontiers in Education Conference, Washington, D.C., USA.

Dijkstra, E. W. (1989). On the Cruelty of Really Teaching Computing Science. *Communications of the ACM, 32*(12), 1398-1404.

Dolmans, D. H. J. M., & Schmidt, H. G. (2006). What Do We Know About Cognitive and Motivational Effects of Small Group Tutorials in Problem-Based Learning? *Advances in Health Sciences Education, 11*(4), 321-336.

Doody, J. (2009). *A Longitudinal Evaluation of the Impact of a Problem-Based Learning Approach to the Teaching of Software Development in Higher Education.* (Doctoral Thesis), Durham University, U.K.

Eckerdal, A., McCartney, R., Moström, J. E., Ratcliffe, M., Sanders, K., & Zander, C. (2006). Putting threshold concepts into context in computer science education. *ACM SIGCSE Bulletin, 38*(3), 103-107.

Eisenstaedt, R. S., Barry, W. E., & Glanz, K. (1990). Problem-based learning: Cognitive retention and cohort traits of randomly selected participants and decliners. *Academic Medicine, 65*(9), 11–12.

Ellis, S., & Dick, P. (2000). *Introduction to Organisational Behaviour* (3rd ed.). London: McGraw-Hill.

Entwistle, N. J. (1997). The Approaches and Study Skills Inventory for Students (ASSIST) *Centre for Research on Learning and Instruction, University of Edinburgh, Edinburgh*.

Entwistle, N. J., Tait, H., & McCune, V. (2000). Patterns of Response to an Approaches to Studying Inventory across Contrasting Groups and Contexts. *European Journal of Psychology of Education, 15*(1), 33-48.

Fincher, S. (1999). *What are we doing when we teach programming?* Paper presented at the 29th Annual Frontiers in Education Conference, 1999. FIE '99., San Juan, Puerto Rico.

Fincher, S., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., . . . Tutty, J. (2005). Programmed to succeed?: a multi-national, multi-institutional study of introductory programming courses *Computing Laboratory Technical Report 1* (Vol. 5, pp. 66).

Hendry, G. D., Frommer, M., & Walker, R. A. (1999). Constructivism and problem-based learning. *Journal of further and higher education, 23*(3), 369-371.

Hidi, S., & Harackiewicz, J. M. (2000). Motivating the academically unmotivated: A critical issue for the 21st century. *Review of educational research, 70*(2), 151-179.

Hmelo-Silver, C. E. (2004). Problem-Based Learning: What and How Do Students Learn? *Educational Psychology Review, 16*(3), 235-266.

Jackson, M. (2003). Why software writing is difficult and will remain so. *Information Processing Letters, 88*(1-2), 13 - 25.

Jenkins, T. (2001). *The motivation of students of programming.* Paper presented at the 6th annual conference on Innovation and technology in computer science education, Canterbury, UK.

Jenkins, T. (2002, August 27- 29). *On the difficulty of learning to program.* Paper presented at the 3rd Annual conference of the LTSN Centre for Information and Computer Sciences, Loughborough, UK.

Kaufman, A., Mennin, S., Waterman, R., Duban, S., Hansbarger, C., Silverblatt, H., . . . Wiese, W. (1989). The New Mexico experiment: Educational innovation and institutional change. *Academic Medicine, 64*(6), 285–294.

Kaufman, D. M. (1995). Preparing faculty as tutors in problem-based learning. *Teaching Improvement Practices—Successful Strategies for Higher Education (eds. W. A. Wright & Associates)*, 101-125.

Kinnunen, P., & Malmi, L. (2005). Problems in Problem-Based Learning--Experiences, Analysis and Lessons Learned on an Introductory Programming Course. *Informatics in Education, 4*(2), 193-214.

Lewis, K. E., & Tamblyn, R. M. (1987). The problem-based learning approach in Baccalaureate nursing education: How effective is it? *Nursing Papers, 19*(2), 17–26.

Lohr, S. (2001). *Go To: The story of the math majors, bridge players, engineers, chess wizards, maverick scientists and iconoclasts - the programmers who created the software revolution.* New York. USA.: Basic Books.

Long, W. F. (2003). Dissonance Detected by Cluster Analysis of Responses to the Approaches and Study Skills Inventory for Students. *Studies in Higher Education, 28*(1), 21-35.

Mamone, S. (1992). Empirical study of motivation in a entry level programming course. *ACM SIGPLAN Notices, 27*(3), 54-60.

Marsh, H. W. (1987). Student's evaluations of university teaching: research findings, methodological issues, and directions for future research. *International Journal of Educational Research, 11*(3), 253-388.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., . . . Wilusz, T. (2001). A Multi-National, Multi-Institutional study of Assessment of Programming Skills of First Year CS Students. *ACM SIGCSE Bulletin, 33*(4), 125-140.

Newble, D., & Clarke, R. (1986). The approaches to learning of students in a traditional and in an innovative problem-based medical school. *Medical Education, 20*(4), 267-273.

Newman, M. (2002). Software Errors Cost U.S. Economy $59.5 Billion Annually. Retrieved 27 August 2009, 2009, from http://www.nist.gov/public_affairs/releases/n02-10.htm

Newman, M. (2004). A pilot systematic review and meta-analysis on the effectiveness of Problem Based Learning. Newcastle upon Tyne: The

Learning and Teaching Support Network subject centre for Medicine, Dentistry and Veterinary Medicine.

Newman, M. (2004). A pilot systematic review and meta-analysis on the effectiveness of Problem Based Learning (pp. 73): The Learning and Teaching Support Network subject centre for Medicine, Dentistry and Veterinary Medicine.

Norman, G. R., & Schmidt, H. G. (1992). The psychological basis of problem-based learning: a review of the evidence. *Academic Medicine*(67), 557-565.

O'Kelly, J. (2005). Designing a hybrid problem-based learning (PBL) course: A case study of first year computer science in NUI Maynooth. In T. Barrett, I. Mac Labhrainn, & H. Fallon (Eds.), *Handbook of Enquiry and Problem-based Learning: Irish Case studies and International Perspectives*.

O'Kelly, J., Mooney, A., Ghent, J., Gaughran, P., Dunne, S., & Bergin, S. (2004). *An Overview of the Integration of Problem Based Learning into an existing Computer Science Programming Module.* Paper presented at the Problem-Based Learning International. Conference 2004: Pleasure by Learning, 2004, Cancun, Mexico.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *JOURNAL OF EDUCATIONAL COMPUTING RESEARCH, 19*(4), 367-381.

Richardson, J. T. E. (2005). The future of research in problem-based learning. In H. Crabtree, A. Darvill, K. Holland, S. MacKay, M. McLoughlin, D. Oakley, & J. Supyk (Eds.), *Problem-based Learning 2004 A Quality Experience?* (pp. 41-59): University of Salford.

Ryan, R. M., & Connell, J. P. (1989). Perceived locus of causality and internalization: Examining reasons for acting in two domains. *Journal of Personality and Social Psychology, 57*(5), 749-761.

Sadlo, G. (1997). Problem-based learning enhances the educational experiences of occupational therapy students. *Education for Health, 10*(1), 101–114.

Sadlo, G., & Richardson, J. T. E. (2003). Approaches to Studying and Perceptions of the Academic Environment in Students Following Problem-Based and Subject-Based Curricula. *Higher Education Research & Development, 22*(3), 253-274.

Schmidt, H. G. (1990). Innovative and conventional curricula compared: What can be said about their effects? In Z. H. Nooman, H. G. Schmidt, & E. S. Ezzat (Eds.), *Innovation in medical education: An evaluation of its present status* (pp. 1-7). New York: Springer.

Schmidt, H. G., Loyens, S. M. M., van Gog, T., & Paas, F. (2007). Problem-Based Learning is Compatible with Human Cognitive Architecture: Commentary on Kirschner, Sweller, and Clark (2006). *EDUCATIONAL PSYCHOLOGIST, 42*(2), 91-97.

Tait, H., & Entwistle, N. J. (1996). Identifying students at risk through ineffective study strategies. *Higher Education, 31*(1), 97-116.

van Grinsven, L., & Tillema, H. (2006). Learning opportunities to support student self-regulation: comparing different instructional formats. *Educational Research, 48*(1), 77 - 91.

Vernon, D. T., & Blake, R. L. (1993). Does problem-based learning work? A meta-analysis of evaluative research. *Academic Medicine, 68*, 550-563.

Waite, W. M., Jackson, M. H., & Diwan, A. (2003). *The conversational classroom.* Paper presented at the The 34th SIGCSE technical symposium on Computer science education, Reno, Navada, USA.

Wiedenbeck, S., LaBelle, D., & Kain, V. N. R. (2004, April). *Factors affecting course outcomes in introductory programming.* Paper presented at the 16th Annual Workshop of the Psychology of Programming Interest Group, Carlow, Ireland.

Wilkerson, L., & Feletti, G. (1989). Problem-based learning: One approach to increasing student participation. The Department Chairperson's Role in Enhancing College Teaching. *New Directions for Teaching and Learning*(37), 51-60.

Williams, G. C., & Deci, E. L. (2007, 27-Nov-2006). Learning Self-Regulation Questionnaire (SRQ-L). Retrieved 29 January, 2009, from http://www.psych.rochester.edu/SDT/measures/selfreg_lrn.html

Woods, D. R. (1996). *Problem-based Learning: How to gain the most from PBL.* Ontario, Canada: Waterdown.

Woods, D. R., Hall, F. L., Eyles, C. H., Hrymak, A. N., & Duncan-Hewitt, W. C. (1996). Tutored versus tutorless groups in problem-based learning. *AMERICAN JOURNAL OF PHARMACEUTICAL EDUCATION, 60*, 231-238.