Chapter 18
Incorporating a Self-Directed Learning Pedagogy in the Computing Classroom:
Problem-Based Learning as a Means to Improving Software Engineering Learning Outcomes

Oisín Cawley
The National College of Ireland, Ireland

Stephan Weibelzahl
The National College of Ireland, Ireland

Ita Richardson
University of Limerick, Ireland

Yvonne Delaney
University of Limerick, Ireland

ABSTRACT

With a focus on addressing the perceived skills gap in Software Engineering (SE) graduates, some educators have looked to employing alternative teaching and learning strategies in the classroom. One such pedagogy is Problem-Based Learning (PBL), an approach the authors have incorporated into the SE curriculum in two separate third-level institutions in Ireland, namely the University of Limerick (UL) and the National College of Ireland (NCI). PBL is an approach to teaching and learning which is quite different to the more typical "lecture" style found in most 3rd level institutions. PBL allows lecturers to meet educational and industry-specific objectives; however, while it has been used widely in Medical and Business schools, its use has not been so widespread with computing educators. PBL is not without its difficulties given that it requires significant changes in the role of the lecturer and the active participation of the students. Here, the authors present the approach taken to implement PBL into their respective

DOI: 10.4018/978-1-4666-5800-4.ch018

programs. They present the pitfalls and obstacles that needed to be addressed, the levels of success that have been achieved so far, and briefly discuss some of the important aspects that Software Engineering lecturers should consider.

INTRODUCTION

Where is the engineering in software engineering (SE)? While there are many technical skills required in the analysis, design, development and implementation of software systems, ask an IT professional to characterize their profession, and you might just as likely solicit the response that they see themselves as an artist, as opposed to a scientist. Given that there is undoubtedly an important design (some might even say creative) element within the practice of SE, it would be reasonable to expect that our SE graduates are also supported in developing non-technical skills.

In addition, if we look at what the academic world has defined under the banner of SE, we clearly see the necessity to arm our graduates with many non-technical skills. Wasserman's eight notions (Wasserman, 1996), for example, include a software process element, which is fundamental for an effective discipline of SE. This software process element focuses on quality through the organization and discipline within the various SE activities. The Software Engineering Body Of Knowledge (SWEBOK)[1] is currently adding an additional knowledge area, titled "Software Engineering Professional Practice," which includes "… subareas of professionalism, group dynamics and psychology, and communication skills." Clearly there is a growing understanding within academia that such "softer" skills play an increasingly important role in the successful outcome of SE projects.

The experiences of two of the authors bears witness to a lot of what has been identified above. OC and IR spent 25 years between them working on SE projects, large and small, in both small and multi-national companies. Their experiences have shown that while technical knowledge is a requirement for much of the SE life cycle, other non-technical skills had been seen to be increasingly important as systems grew in complexity and the business functions became less tolerant with overdue and over-budget projects. Systems complexity, in this sense, is not only a technical concern but also relates to the change in team dynamics as the number of stakeholders and project participants increase. This type of complexity requires oral, written, interpersonal and team working skills that some authors argue our graduates are not being adequately equipped in when compared to their technical abilities (Davies, 2000; Cotton, 1993; Connor and Shaw, 2008). We have recognized that, when using Problem-based learning in our classes, we can provide students with these technical and non-technical skills.

WHAT IS PROBLEM-BASED LEARNING?

"Problem-based learning (PBL) is apprenticeship for real-life problem solving, helping students acquire the knowledge and skills required in the workplace" (Dunlap, 2005). PBL has a long "intellectual history" with its origins in the "philosophies of rationalism and American functionalism" (Dewey, 1929; Schmidt, 1993). Current day PBL emerged in the 1950's and 1960's in Case Western Reserve University and McMaster University respectively (Prince & Felder, 2006). In the late sixties, Howard Barrows joined the faculty at McMaster University in Canada. During that time he collaborated with others and developed the approach to learning now called Problem-based Learning (Schmidt & De Volder, 1984). By the early seventies, Problem-based Learning was installed as a total approach to learning and