

# On-line Learning Development: an incremental approach

Jon M Pearce  
Department of Information Systems  
University of Melbourne, Australia

Michelle K Livett  
School of Physics  
University of Melbourne, Australia

## Abstract

During the past five years the authors have been involved in several funded development projects involving the use of computers in tertiary teaching (primarily in the physics discipline). The current state of those projects covers the full range from dead, still under development, to successful usage by students.

The nature of these projects also covers a wide range. Some focussed on the production of a significant piece of software to be incorporated into a teaching programme. Others involved an "incremental" approach to courseware design: they were web-based and could be implemented and extended even though they weren't yet in a completed state. Each project was initially funded with about \$50,000 (by CAUT, CUTSD, the University of Melbourne, and, in a smaller way, AUDF). Although the "success" of such projects can be measured in many ways—learning outcomes, acceptance by peers, integration into courses, completion of original aims, etc.—it appears that success as indicated by at least one of these measures was determined in part by the nature of the project.

Those projects focussing on the production of a significant piece of software usually relied on, in part, outside programming assistance. They involved sophisticated, highly interactive software necessitating them reaching an advanced state of development—near completion—before being implemented. Their success was sensitive to their acceptance by peers, ease of integration into a course, and students overcoming a "learning threshold". There was a significant risk of failure before completion.

The Web-based projects, however, offered a far more flexible approach to development. They still required the development of a number of highly interactive components, but required a lower threshold to initiate, and stages could be implemented even though later stages were not yet complete. Due to the "soft" programming requirements of many of these components (*QuickTime* editing, *Flash* animations, implementing third party Java applets) much of the work could be done by academics rather than by programmers. Given the nature of these particular projects, it was also true that more of the development work *had* to be done by the academics as the pedagogy was so tightly integrated with the content that it was not appropriate to outsource much of the development.

It was the experience of the authors that this second style of project, incremental in nature, better suited the work patterns of busy academics and was more likely to be successfully implemented into a teaching programme. The tight integration of project design and development with the content and pedagogy of the subject discipline meant that it was less suitable to contract out most of the production work to third parties. Hence the time commitment by the academics was considerable, but the resulting resources targeted conceptual difficulties and were highly valued by students.

This paper presents several of these projects and discusses the nature of each, its current status, the development process, the interactive technologies used and the information gained from student trials. From this review of projects, various conclusions and pitfalls can be identified about the development processes being undertaken in an academic environment.