Chapter II
Constructive Alignment in SE Education: Aligning to What?

Jocelyn Armarego
Murdoch University, Western Australia

ABSTRACT

Practitioner studies suggest that formal IT-related education is not developing the skills and knowledge needed by graduates in daily work. In particular, a shift in focus from technical competency to the soft and metacognitive skills is identified. This chapter argues that a framework for learning can be developed that more closely models the experiences of practitioners, and addresses their expectations of novice software engineers. Evaluation of a study incorporating three action research cycles shows that what is needed is a mapping between the characteristics of professional practice and the learning model that is applied. The research shows that a relationship also exists between learner and learning model, and that this relationship can be exploited in the development of competent discipline practitioners.

INTRODUCTION

In the late 1960s those involved in the development of software agreed that one mechanism for dealing with intrinsic difficulties (eg complexity, (in)visibility, and changeability (Brooks, 1986)) was to embed its production within an applied science environment. Royce (1970) was the first to note explicitly that an engineering approach was required. The implication of this alignment was that, like other engineering endeavours, methods, tools and procedures must be applied in a systematic way to contribute to the overall purpose of the process, control it and enable the development of a quality product.

This interest in engineering is mirrored in the education of software developers, with initially an exponential growth in offerings of undergraduate software degrees within an engineering environment. Increasingly, education for software development focuses on process and repeatability, modelling scientific and engineering methodologies. The underlying assumption of this approach is that 'good' software development is achieved by applying scientific investigative techniques (Pfleeger, 1999).

Practitioner-based studies (eg., Trauth, Farwell, & Lee, 1993; Lethbridge, 2000; Lee, 2004) assist us in building a profile of a practicing IT professional. The synthesis of these is that the skills and knowledge required to be active as competent practitioners are multidisciplinary: industry requires professionals who integrate into the organisational structure, and, rather than cope specifically with today's perceived problems, have models, skills and analytical techniques that allow them to evaluate and apply appropriate emerging technologies and to manage the process of delivering solutions. More broadly, software technology is seen as a rapidly shifting landscape: new methods, tools, platforms, user expectations, and software markets underscore the need for education that provides professionals with the ability to adapt quickly.

Developing Education-Learner-Practitioner Alignments[1]

Freed (1992) coined the term 'relentless innovation' to describe the capacity to invent and implement new ideas that will impact on every facet of life. Oliver (2000) suggested the rate of innovation is so prolific that most of the knowledge which will be used by the end of the first decade of the twenty-first century has yet to be invented. The speed with which technology evolves, the multiplicity of its impact on society and the ramifications of that impact mean that metacognitive and knowledge construction skills as well as adaptability become vital for professionals working with technology. Professional practitioners with such skills become *agents of change* (Garlan, Gluch, & Tomayko, 1997).

However, the basic features of most engineering training programmes have hardly been challenged since engineering schools were established (Mulder, 2006). In general this education is based on a normative professional education curriculum, in which students first study basic science, then the relevant applied science (Waks, 2001), so that learning may be viewed as a progression to expertise through task analysis, strategy selection, try-out and repetition (Winn & Snyder, 1996). The risk is that strict adherence to engineering and science methodologies hampers the quintessential creativity of the design process for software (Lubars, Potts, & Richer, 1993; Maiden & Gizikis, 2001; Maiden & Sutcliffe, 1992; Thomas, Lee, & Danis, 2002).

The aim of this chapter therefore is to explore the degree of alignment between the actuality of practice in the discipline and the models of learning provided in formal education for software development. An overview of both the dominant pedagogy for formal education in IT disciplines, and practitioner studies undertaken over the last 15 years establishes a base for this exploration.

An Action Research project, undertaken within Murdoch University's Software Engineering (SE) programme, provided the context for developing a model for alignment between formal education for SE and industry requirements. In order to achieve this, several techniques, including curriculum mapping and discipline decoding, were applied during the project to establish and then evaluate the alignments identified. The chapter continues by exploring the importance of alignment between student and learning environment, so that the eventual outcome, affinity between discipline, learning environment and graduate practitioner may be achieved.

CONTEXT

The context for the Action Research[2] project was the SE programme within the School of Engineer-