Software Engineering: Effective Teaching and Learning Approaches and Practices

Heidi J. C. Ellis  
*Trinity College, USA*

Steven A. Demurjian  
*University of Connecticut, USA*

J. Fernando Naveda  
*Rochester Institute of Technology, USA*
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 (e.g., 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

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 project was the SE programme within the School of Engineer-
Constructive Alignment in SE Education

In an attempt to align the characteristics of the discipline with appropriate learning environments, and to address knowledge gaps identified by practitioners, interventions based on different learning models were embedded in the curriculum over three cycles:

- **Cycle 1**: The Cognitive Apprenticeship (Brown, Collins, & Duguid, 1989) model as a mechanism for enabling authentic learning and facilitating knowledge transfer.
- **Cycle 2**: Problem-based learning (PBL) (Barrows & Tamblyn, 1980) as the basis for a model that focuses on students dealing with ill-structured problems by taking control of their learning. The model developed and applied in this cycle also addresses issues of enabling creativity within a supportive learning environment (Armarego, 2005).
- **Cycle 3**: A hybrid model developed on the basis of reflection on the interventions of the previous two cycles. Based on the constructivist paradigm, this Studio Learning model exploits the reflective practitioner (Schön, 1983) concept of professional learning by incorporating some elements of Cognitive Apprenticeship with components of problem-based learning and creativity-enhancing strategies. The focus is on the longer-term success of the learning strategies identified as appropriate for SE education (Armarego, 2007a; Armarego & Fowler, 2005).

The SE curriculum at Murdoch is an integrated one – all courses are prescribed, therefore a very precise understanding of what knowledge students have constructed is available. As Armarego (2002) indicates, initial changes were made only to the ‘capstone’ course. However, issues identified in the evaluation (see Armarego, 2004) indicated changes were required earlier in the curriculum. Cycles 1 and 2 of the project addressed this aspect by focussing on changing student perception of ‘appropriate’ learning of SE. Cycle 3 consolidated the evolved learning model and extended it, not only to all SE learning within the curriculum, but to the final years of all engineering learning (Armarego & Fowler, 2005) in the School.

**CHARACTERISTICS OF THE DISCIPLINE**

**The Engineering of Software**

The alignment of software development with science and engineering has been seen as a means to leverage from the ‘status’ of these domains: the profession of scientifically trained engineer came into existence in the 18th and 19th centuries as a product of the Enlightenment. For engineers it meant rethinking traditional technologies in order to rationalise and optimise them. However, Mulder (2006) notes that engineers sometimes failed to recognise that the issue at stake was not always a scientifically-/mathematically-solvable optimisation problem, but a choice between irreconcilable norms and values.

The implication of the alignment of software development with science and engineering is 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.

However, by the late 1960s philosophers such as Habermas (1972) criticised the ideological character of science-based technology – successful technologies were seen to challenge society and affect it as a whole. A deep understanding of the motives and desires of people who would be relating to the new technology, developed through interaction, was critical.

**The Crafting of Software**

Software development has also been described as a ‘craft’. The negative connotations of this label...
include an inability to consistently guarantee a quality product, fit for the purpose for which it was developed, produced on time and within budget. The rates of successful projects reported in the mid 1990s are not significantly higher than those reported in the 1970s and 1980s (Mann, 1996), and continue to be low in the 2000s.

There are positive implications as well for the label ‘craft’. Each system is considered a unique synergy between the hardware, software and organisational context in which it will be used. This viewpoint suggests that the development process cannot be repeatable, as the forces at play will differ for each context: continually changing as understanding of the characteristics of the developing system grows in all stakeholders. From this perspective software is a collaborative invention. Its development is an exploratory and self-correcting dialogue (Bach, 1999), based on insight-driven knowledge discovery (Guindon, 1989) facilitated by opportunistic behaviour (Guindon, 1990; Visser, 1992).

**EDUCATION FOR THE DISCIPLINE**

Hannafin (1997) and Reeves (1994) suggest that several dimensions are relevant in the description of learning systems:

- **Epistemological foundations**: Are concerned with theories about the nature of knowledge, and describe the world view to be disseminated. At one extreme (objectivism), content aims to be comprehensive and accurate, and based on advice from experts in the field. At the other (constructivism), content reflects the spectrum of views in the domain, providing multiple perspectives/options for constructing knowledge.

- **Psychological foundations**: Represent beliefs about how individuals think and learn. On this continuum, shaping desirable behaviours via stimuli, feedback, reinforcement etc at one pole contrasts with a cognitivist emphasis on mental models and the connections between them. The type of knowledge to be constructed is seen to drive the learning strategy employed.

- **Philosophical foundations**: Emphasise how to-be-learned domains are represented and affordances provided to support learning. An instructivist foundation stresses the importance of goals and objectives drawn from the domain. Constructivist foundations, on the other hand, stress the primacy of learner intentions, experience and metacognitive strategies through a rich environment that can be tailored to individual needs.

These dimensions describe the nature of learning, the methods and strategies employed, and the ways in which the discipline should be organised and made available to the learner.

Although any software development project is acknowledged as knowledge-intensive, with many concepts developed to ease or guide the processing of knowing (Robillard, 1999), and learning (Klemola & Rilling, 2002), what is actually taught within a discipline is a complex synthesis deriving from the ideology of the discipline, the context of the learning and the ‘tools’ used to facilitate that learning, all, in theory, influenced by the needs of practitioners in the discipline. Figure 1 describes a conceptual framework that identifies the elements of this synthesis: the bodies of knowledge (BoKs) and model curricula are a distillation of expert opinion and domain-specific texts. The breakdown is seen to cover the areas discussed in texts and standards, either identically, or, as noted in the Software Engineering Body of Knowledge (SWEBoK), derived from these and other sources to reflect a consensus and identify mature and stable concepts (Sawyer & Kotonya, 2000) in the discipline.

At the same time, a perspective (composed of the epistemological, psychological and philosophical foundations noted above) also exerts
influence on each of the domain, BoKs etc and theories of learning. Within the IT disciplines this has led to multiple approaches to its definition and study: the work of Iivari (1991) and Glass (1992) identified and categorised these, based on epistemological and ontological positions taken. The implication of this is a different understanding of the discipline and education for it dependent on the stance (perspective) adopted. This poses a serious challenge for the learning of software development practice.

The accepted view, that a science/engineering approach will ensure quality, influences the learning of SE: by implication a scientific/engineering education is seen as the mechanism to train students to be competent practitioners. The same is true outside the science/engineering academic faculties: Benson (2003) notes that within the emerging information systems (IS) discipline of the 1970s, academics were migrants to the discipline, with an overwhelming majority having qualifications in other areas, most often computer science. Practitioners also relied heavily on scientific, mathematic and engineering disciplines, many with engineering and manufacturing backgrounds.

These influences are mirrored in attempts at developing model curricula, with the occasional addition of guidelines addressing generic attributes. Shackelford (2005) provides an overview of what might be considered computing today (the space for SE is illustrated in Figure 2). At a fundamental level, the assumptions made on, for example, the nature of the system or the importance of its context, and the nature of knowledge, influence the perspective taken and how the work is undertaken. However, each of the volumes of the Computing Curriculum (CC-CS (Engel & Roberts, 2001), CC-IS (Gorgone et al., 2002) and CC-SE (LeBlanc & Sobel, 2004)), which help determine the learning situation for a discipline, applies the same model and draws on the same types of sources.

Within the broad IT specialisations in general, the underlying assumption is that the world works rationally and that therefore ‘good’ software development is achieved by applying (from a choice of) scientific investigative techniques. In this positivist approach, borrowing from the physical sciences, software developers build models based on: theoretical and scientific knowledge; engi-
neering knowledge – experiential and including what skills are needed, how tools work together, what has/has not worked in the past; biomedical and epidemiological knowledge – experiential, this captures evidence about causation and social, economic and institutional knowledge – who and what are involved in what we are observing (Pfleeger, 1999). By these means the ‘scientific’ software developer seek relationships that add to an understanding of what makes software good. These are applied to increase the number of times good software is produced, based on a cause-effect search: if s/he can find out what process activities, tools, measurements cause good software s/he can build an effective software process that will produce good software every time (Pfleeger, 1999).

Also applied within IS education such approaches lean towards project management-based methods, techniques and tools, and, while successful in creating a range of artefacts, do not succeed in the development of management information systems (Banks, 2003). Banks concludes that the weakness inherent in approaches which lend themselves to ‘cookbooks’ with clearly defined problems, rigid method and limited range of outcomes but tangible skills in students is the lesser regard for real-world influences and pressures.

Therefore, while a review of major model curricula for software development (ie IS, CS and SE) shows that, in general terms, a graduate should emerge from formal education with knowledge of the basic software development processes (and therefore, in theory be able to produce successful software), this does not acknowledge either the multi-disciplinary skills highlighted by practitioners as missing in formal education or the generic intellectual abilities and skills which, although highly valued by employers, are sometimes given only ‘lip service’ in tertiary education curricula (Bentley, Lowry, & Sandy, 1999).

DISCIPLINE DECODING

One of the primary motivations for the development of models of teaching and learning in which practitioners can be more involved in the research on how people think and how students learn has been a concern with the disciplinary nature of
learning. The result of the decoding process is a model of the skills identified as necessary within a discipline.

Disciplines differ in the strategies and the ‘ways of thinking’ practitioners apply. However, although these are essential for both understanding the discipline and acting within it, they are not usually presented to students explicitly. Parnas and Clements (1986) suggest that, given an irrational design process (ie all design processes), the documentation should make it appear as though it were rational. They justify this faking of the appearance of rationality through the need to make the eventual maintenance task easier, as well as enabling new members of the design team to absorb knowledge about the project more easily. However, as some research (eg., Nguyen & Swatman, 2000) suggests, the process to such simplification is hidden and leads to unreal expectations in novice undertakings. According to Middendorf and Pace (1986), this dichotomy has led to a gap between strategies for learning and the skills necessary in specific disciplines.

Therefore, although practitioner studies agree that the base case of content knowledge is covered in models used in university programmes, a closer look reveals the depth of the mismatch between practitioner needs and formal education for software development.

**Practitioner Perspectives**

In his *Point/Counterpoint* discussion, Bach (1997) stated that one reason SE is not more seriously studied is the common industry belief that most of the books and classes that teach it are impractical. An overview of the studies undertaken to gain a practitioner perspective indicates that such an indictment is not too far from the mark.

Most of the studies noted below address the requirements for software development activities by examining the general importance of specific topics, as perceived by different stakeholders. Since different approaches are taken in gaining this knowledge from different target groups: surveys, focus groups, fora or interviews applied to experienced practitioners, managers, recruitment staff, students and recent graduates, as well as examination of job advertisements over the disciplines of IS, CS and Engineering, some insight into the practitioner perspective is possible.

In IS practitioner studies since the early 1990s (eg., Trauth et al., 1993; Parker et al., 1999; Lee, 2004) a long term shift from programming and other technical subjects to business analysis and people-oriented skills is significant – a change in emphasis to both generic attributes and managerial knowledge. From the student perspective, awareness of the need for ‘career resilience’ has surfaced (Waterman, Waterman, & Collard, 1994), while a technology-relevant degree is less necessary. Lee (1999) concluded that academic programmes should emphasise information searching and problem formulation (as opposed to problem solving alone) so that students can deal more effectively with the challenges of industry. He noted that interpersonal communication accounts for the most important means of knowledge transfer in technological work, with team members as the most utilised inter-personal information source.

From a later study Lee found that one of the reality shocks involved in the socialisation of new graduates to work was the onus of teaching themselves what they needed to know in order to perform the task successfully. He concludes...

*educators should also help students to develop their initiatives and abilities to deal with ill-structured problems. This would require approaches which emphasize independent learning and collaborative teamwork.* (Lee, 2004, p 135)

Fewer studies address the skills and knowledge needed in SE and CS. Turley and Bieman (1995) examined professional Software Engineers in an attempt to identify the competencies and demographics that contribute to ‘excellence’ in performance. They provide a set of thirty eight
competencies that express a broad range of behaviours required of an IT professional engaged in the creation of software products (as opposed to maintenance, management etc). They identify four categories of competencies which differentiate between exceptional (XP) and non-exceptional (NXP) performers (see Table 1). Of the statistically significant competencies associated with exceptional performance most are seen to cluster around the theme of external focus, with only Mastery of Skills and Techniques as a self-directed (internal) skill. Earlier Turley (1991) concluded that education needed to support the development of differential skills (namely interpersonal skills and personal attributes) through the creation of learning situations that stress these. Lethbridge (2000) also examined the industry perception: his aim was to gain a practitioner ranking of the usefulness of specific topics compiled from the curricula of (emerging) SCE (Software and Computer Engineering) and CS, the influence of these on respondents’ career and how much they had learned formally compared to what was required as a professional. Of relevance to our consideration, Lethbridge computed overall importance of topics, based on the average of both importance of details and influence. The results of his work indicate the existence of significant gaps between formal learning and importance on the job. Of the top ten topics exhibiting considerable gap, 50% reflect ‘soft’ knowledge (eg negotiation (84% gap), leadership (73%), ethics and professionalism (62%)).

Studies in the Australian context support these findings. Respondents to a study by Scott and Yates (2002) noted that learning profession-specific content provides the ‘scaffold’ for the important task of career-long professional learning: the skills to undertake this are of great importance, with the ability to know when and when not to deploy technical expertise, and how to continuously update it, the keys to successful professional practice.

From Scott and Wilson (2002)’s work, the finding is that, while the successful professional must possess a high level of profession-specific

<table>
<thead>
<tr>
<th>Competency</th>
<th>XP Rank</th>
<th>NXP Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task Accomplishment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mastery of Skills &amp; Techniques</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Personal Attributes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Driven by a desire to contribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perseverance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintains ‘Big Picture’ view</td>
<td></td>
<td>5 (NXP)</td>
</tr>
<tr>
<td>Desire to do/bias for action</td>
<td></td>
<td>XP (XP)</td>
</tr>
<tr>
<td>Driven by a sense of mission</td>
<td></td>
<td>XP (XP)</td>
</tr>
<tr>
<td><em>Exhibits and articulates strong beliefs and convictions</em></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Proactive role with management</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td><strong>Situational Skills</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responds to schedule pressures by sacrificing parts of the design process</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>Interpersonal Skills</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seeks help from others</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Helps others</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Willingness to confront others</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Numbers indicate ranking based on statistical significance results of a t-test. Items not numbered are the result of a discriminant analysis based on Q-sort results. Competency element in italics indicates both tests identify this as significant.
technical expertise, such skills have little value without other skills:

...when the unexpected occurs, what is most telling is being able to tolerate the uncertainty and ambiguity of the situation, having well developed reciprocal networks upon which to call to identify potentially relevant solutions, being able to ‘read’ the total technical and social components of a troubling situation, and then being able to apply a high level of appropriate technical skill in partnership with other team members to resolve the situation. (Scott & Wilson, 2002, p 6)

The synthesis of these studies implies a need to enable students to not only learn to use past experience on a general level, but to also be able to deal with each new problem situation in its own terms, requiring certain generic intellectual abilities and skills. Gott et al (1993) posit that this adaptive/generative capability suggests the performer not only knows the procedural steps for problem solving but also understands when to deploy them and why they work. The implication of this is effort spent on higher (metacognitive) learning skills, including abstraction and reflection. However, merely applying knowledge has been identified as the aim of undergraduate education, so that generally only the lower three (ie foundational) levels of Bloom (1956)’s taxonomy of cognitive learning have been chosen as educational objectives, since they represent

what knowledge may be reasonably learned during an undergraduate education, (Sobel, 2003, p 6),
effectively ignoring the development of higher level skills (analysis, synthesis, evaluation) in formal (undergraduate) education. This runs counter to Thomas et al (2002)’s suggestion of a (critically) widening gap between the degree of flexibility and creativity needed to adapt to a changing world and the capacity to do so.

ALIGNING EDUCATION TO PRACTICE

Reigeluth (1997) argues that the current paradigm of education is based on standardisation, conformity and compliance, geared to the mass production of industrial age manufacture. This does not equate with the needs of the late 20th/early 21st century job market, which revolves around problem solving, teamwork, communications, initiative taking and diverse perspectives. What this implies is a lack of coincidence between the actuality of practice in the discipline and the instructional design supposed to model it – suggesting the need for a new paradigm, based on customisation, diversity and initiative, to suit the needs of the information age.

Felder and Brent (2005) assert that traditional engineering education does little to provide students with the systemic perspective on individual subjects (a global perspective) they need to function effectively, and the ones who take too long to get it by themselves are at risk academically. They see most engineering instruction oriented toward students with specific traits – introverts (favouring lecturing and individual assignments rather than active class involvement and cooperative learning), intuitors (preferring emphasis on science and math fundamentals rather than engineering applications and operations), thinkers (favouring objective analysis rather than interpersonal considerations in decision-making), and judgers (preferring emphasis on following the syllabus and meeting assignment deadlines rather than on exploration of ideas and creative problem solving). Holt and Solomon (1996) point out that, while engineering education relies heavily on problem solving and engineering science, it limits the opportunities of all learners to develop the skills required for proficiency in two key areas: design and invention (requiring a divergent approach), and business management (requiring accommodative skills). The work of Lumsdaine and Lumsdaine (1995) suggests that between 20%
and 40% of student intake to engineering is lost through not catering for students with strengths in communications and team work or creative problem solving, synthesis and design.

In SE, Glass (1995) suggests that discipline and creativity are the odd couple of software development – the discipline imposed by methodology, for example, forms a frame for the opportunistic creativity of design. The educational dilemma becomes one of providing a base that enables software developers to both create and engineer the systems they build: to be adaptable to the changing environment that is inevitable in their chosen discipline. However, criticism has arisen regarding engineering graduates’ ability to be creative (Cropley & Cropley, 1998). The need for flexibility, fluency and originality in day-to-day dealings, which typically define the creative effort (Guilford, 1967), is seen as lacking from their education.

The inadequacy of formal education in training competent practitioners, then, may be partly explained by the ‘incorrect’ learning environment that results from the poor fit between the characteristics of the discipline identified by practitioners and those of the learning model. A solution can be proposed through the development of a new framework for SE education. This framework should:

- Be based on constructivist theory (as more suitable for learning in domains involving ill-structured problems (Spiro, Feltovich, Jacobson, & Coulson, 1991) with a focus on strategic knowledge to enhance knowledge construction and transfer. This includes metacognitive strategies for directing, monitoring and evaluating learning.
- Be placed within a situated experiential learning environment where authenticity (with rich contextual information) is exploited (Dreyfus & Dreyfus, 1986). Focusing on the solution of authentic problems as a context for learning provides students with entry to the community of practice to which they will belong.
- Provide the student with a learning environment that has an emphasis on modelling practice, making tacit knowledge explicit and thus empowering students to think independently.

Several learning models apply these concepts. As noted previously, the project looked specifically at Cognitive Apprenticeship and problem-based learning as exemplars. However, there is a suggestion in the literature that efforts to help students learn at Bloom’s higher-order levels may be impeded by a mismatch between the kinds of thinking actually required in specific disciplines and generic formulae for encouraging higher-order thinking (Middendorf & Pace, 1986). In the final analysis, applying generic learning models (even non-traditional ones) for situated, higher-order learning that is student-centred may run counter to an important strand in the current thinking about teaching. This stresses the disciplinary nature of knowledge. As a tool for learning, the model must be adapted to the discipline. The development of a curriculum map aligns the needs of the discipline with the educational strategies to address these concerns.

**Curriculum Mapping for Constructive Alignment**

As both curriculum development and learning theory move away from behavioural to cognitivist and constructivist approaches in order to address the needs of both the discipline and changing context for the discipline, the value of alignment is enhanced.

The basis of a framework for a learning environment is a ‘constructive alignment’ (Biggs, 1999) of objectives, teaching context and assessment tasks. Based on the discussions of Brown, Bull, and Pendlebury (1997), aligning these components achieves the following aims: the
Constructive Alignment in SE Education

educational expectation (learning objective) is mapped to learning activities likely to achieve these (teaching context) while assessment tasks focus on the quality of the learning process. A model of alignment, based on the work of the engineering subject centre of the learning and teaching support network (LTSN, 2002), was applied within the research project (see Figure 3).

In order to facilitate all the alignments required, a map of the curriculum for SE at Murdoch University was constructed. Curriculum mapping, as an evaluative tool attributed to English (1978), has been used primarily in schools, with limited use in higher education. English advocated the use of mapping to ensure that the constructive alignment described above - alignment of declaration, delivery, learning and assessment of individual skills - is achieved.

The outcome of these initial phases, examining curricula and learning models, and decoding the discipline through a meta-analysis of practitioner perspectives, was to confirm the need to build into the curriculum a focus on generic and soft skills as part of the outcomes of each course within the programme, to address both practitioner and discipline needs. To maximise effectiveness, these had to be embedded into the knowledge base constructed by the students during their learning. This has the advantage of enabling students to develop the requisite skills situated within the learning context but, of course, required extensive adaptation of the existing learning environment.

Within the project undertaken, curriculum mapping was tackled course by course, commencing with the initial SE course offered (identified as ENG260), which addresses Requirements Engineering. This was categorised firstly by the broad area of curriculum and then by the learning outcomes to be addressed. The map was based on scrutiny of documentation related to the course; in particular syllabus and course outline information provided to students at the commencement of the semester. These detail topics to be covered, assessment elements and criteria and expected demonstrable outcomes. The data gleaned from all of these were initially mapped to Murdoch’s generic graduate outcomes, and then, as progress was made in developing the activities to address the learning outcomes identified, to these as well. Figure 4 shows the mapping necessary for alignment. The Learning Objectives are determined from the appropriate BoKs and model curricula, tempered by our understanding of the needs of

Figure 3. Alignment between outcomes and assessment (adapted from LTSN, 2002)
Constructive Alignment in SE Education

practitioners in our context. The topics addressed (indicated as Domain) are mapped to Murdoch’s Graduate Attributes. The Problem(s) identify the activity that will address these objectives. Because the course has been presented within a PBL environment (and hence problem-driven) these are never lectures nor simply assessment items or tutorial/laboratory exercises. Students engage with the required content through identifying, exploring and subsequently solving specific problem scenarios. These scenarios are exposed progressively by means of triggers (Figure 5 is one example – at this point students have no prior knowledge of SE estimation techniques).

Figure 4. (excerpt from) Learning objectives - ENG260

<table>
<thead>
<tr>
<th>Objective</th>
<th>Domain</th>
<th>Graduate Attributes</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>To incrementally build knowledge on:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Requirements Engineering</td>
<td>Concepts</td>
<td>Professional Knowledge</td>
<td>1, 2, 6</td>
</tr>
<tr>
<td>2 Elements of the SDLC, both classical and object-oriented</td>
<td>Requirements Engineering Process</td>
<td>Professional Knowledge</td>
<td>2, 6</td>
</tr>
<tr>
<td>To identify:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Components of Requirements Specification</td>
<td>Concepts</td>
<td>Professional Knowledge</td>
<td>2</td>
</tr>
<tr>
<td>4 SDLC Process Models (classical and OO)</td>
<td>Requirements Engineering Process</td>
<td>Professional Knowledge</td>
<td>6, 2</td>
</tr>
<tr>
<td>To be aware of:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Group Dynamics and collaboration in the Software Development process (Team and Stakeholder)</td>
<td>People Issues/Teamwork</td>
<td>Inter-disciplinality/Social Interaction</td>
<td>3 +</td>
</tr>
<tr>
<td>6 Historical issues in Software Development</td>
<td>History</td>
<td>Professional Knowledge</td>
<td>2 +</td>
</tr>
<tr>
<td>To develop skills in:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Creating and evaluating deliverables of the Requirements Phase (complementary models, documentation)</td>
<td>Requirements Engineering Process</td>
<td>Analysis &amp; Problem Solving/Communications</td>
<td>2 +</td>
</tr>
<tr>
<td>8 Group Collaboration</td>
<td>People Issues/Teamwork</td>
<td>Social Interaction</td>
<td>3</td>
</tr>
</tbody>
</table>

26
Curriculum mapping may therefore be considered a traceability exercise: each ‘requirement’ (learning objective) is designed for (triggering one or more problem component/learning object) and may lead to an artefact (an assessment element). The appropriate learning environment is determined by the ‘fit’ of all components to the course and ultimately the overall programme (thus placing emphasis on alignment of elements identified in Figure 1 with those in Figure 3).

It should be noted that the development of the learning environment was continuing throughout the project: the initial model – based on Cognitive Apprenticeship, evolved to a model based on PBL (CreativePBL) and finally to Studio Learning. As Figure 3 indicates, alignment feedback informs the refining of the intended learning outcomes, and hence the learning activities, for subsequent offerings of the course. In this context, ongoing project evaluation indicated the process-oriented approach advocated in PBL acted as an alignment inhibitor by reinforcing the perception of RE is a smooth process of sequential stages – the contingency measures advocated by Andresen, Boud, and Cohen (1995) as needing to be available in the creative nature of design, could not be easily incorporated.

A learning model based on the ‘studio’ approach (itself modelled on the 19th century atelier-based training at the Parisian Ecole des Beaux-Arts), that also emphasised the development of reflective skills and sensibilities (Schön, 1983) was implemented as the learning environment of choice. This Studio Learning model incorporates some elements of Cognitive Apprenticeship with components of problem-based learning and creativity-enhancing strategies. The model supports the idea that learning is defined in terms of dynamic sets of relationships whose interactions and interdependencies create and control conditions that are supportive of specified concepts within a discipline.

Developing a Student-Education Alignment

Student approaches to both learning and the learning environment can be investigated through several diagnostic instruments. Within the study, learning styles (Kolb, 1984; Soloman & Felder, 1999), temperament (Keirsey & Bates, 1984), study approaches (Entwistle & Ramsden, 1983) and relationship to learning activities (Meyer & Boulton-Lewis, 1997) were all incorporated. The results of these instruments help build several profiles of the student cohorts. Important in this context was the individual learning styles and individual approaches to learning. The results confirmed other research (Entwistle & Tait, 1990, 1995; Tynjälä, Salminen, Sutela, Nuutinen, & Pitkänen, 2005) about students with specific learning styles having a preference for surface
learning and ‘being taught’, and indicated that students’ conceptions of the characteristics of their learning environments were related to their study orientations and strategies.

Other research within this School (Armarego, Fowler, & Roy, 2001) indicates that engineering students’ motivation and success can be adversely affected if their learning styles, and the learning styles of the staff teaching them, are not taken into account. There is considerable evidence that a mismatch, between lecturers’ expectations of the way students learn and students’ own individual preferred learning styles, disadvantage students. Research suggests that these mismatches lead to lack of motivation and interest in students and affect their success (Felder, 1996; UWA, 1996; Zywno & Waalen, 2001).

These findings were supported by the project discussed in this chapter, strengthening indications of the importance of additional alignments – teacher and learning environment to student. Learning styles instruments, when applied to engineering academic staff, also indicated a strong Converger approach to teaching. The implication of this was that the dominant teaching style did not exhibit the adaptability and flexibility required by either the characteristics of the discipline or the learning environment being developed.

The term constructive alignment, therefore, goes beyond the need to ensure that teaching, assessment and every aspect of the teaching-learning environment are aligned to the main aims or intended learning outcomes of a course. When the course is not aligned with learner interests or the situation constrains the student’s approach to learning, the dependent learner mode will tend to dominate – control of the learning process is relinquished to the teacher, while the student will demand carefully articulated structure, clear guidance and clearly-defined assessment (Armarego, 2007b). A dependent learner, therefore, does not align with the discipline characteristics described earlier in this chapter. Staff development, to introduce experiential learning models and ‘teaching around the learning cycle’ (Felder, 1996) are advocated (Armarego & Fowler (2005) also discusses the staff development implemented in this project).

CONCEPTUAL MODEL OF ALIGNMENT

The result of the investigation described here, and the Action Research project that underpins it, is the development of a complex model that aligns discipline competencies with student characteristics with learning environment, as illustrated in Figure 6.

This chapter argues that traditional formal education does not meet the competency expectations of industry. Practitioner dissatisfaction with formal education focuses on non-technical components of competency: they look for graduates who are flexible, adaptable in the organisational environment and can continue learning. These have been identified as cognitive skills related to higher order learning, strategies to enable opportunism and creativity and the development of emotional intelligence.

The three cycles of this project explored alternative learning models to evaluate their appropriateness for addressing these issues. A shift in focus from technical competency to the soft and metacognitive skills that enable the competent practice of SE was achieved. Each intervention strategy addressed specific concerns and, through evaluation of and reflection on the intervention, strategies are refined for the next cycle to address additional issues identified:

- **Cycle 1 – Cognitive Apprenticeship**: Focus on authenticity and transfer of skills acquired to other courses and, eventually, to the profession. This cycle highlighted student problems in generalising their learning, and in willingness to apply previous knowledge to the ‘new’ learning. In effect they were
Constructive Alignment in SE Education

A significant finding of this cycle related to student emphasis on ‘correct’ answers to problem solving undertaken. Students focussed on learning the tools and techniques of SE at the expense of a broader (and more abstract) understanding within the discipline.

- **Cycle 2 – CreativePBL**: Focus on student-centred learning; creativity and adaptability. This model was developed to address the deficiencies of the Apprenticeship model that were identified in Cycle 1. It was developed to focus on creativity and divergent thinking, so that, instead of students aimed at finding the single, best, correct answer to a standard problem in the shortest time (convergent thinking) they aimed at redefining or discovering problems and solving them by means of branching out, making unexpected associations, applying the known in unusual ways, or seeing unexpected implications. However, process itself acted as a deterrent to student motivation to study and to exploit the creativity being nurtured – opportunism was difficult within the process and hence flexibility inhibited; here a focus on process detracted from the ‘authenticity’ of the environment.

- **Cycle 3 – Studio Learning**: Focus on deep learning; opportunism and metalearning. This model was developed to gain leverage from the positive elements of the models previously applied. Here the strategy was to reach all types of learners by ‘teaching around the cycle’6, thus enabling students to develop the mental dexterity required in professional practice, and introducing the importance of contingency measures and opportunistic creativity. The Studio environment also provided the opportunity for students to adopt expert strategies – the teacher acts as guide or ‘consultant’ in these processes and helps students to reflect critically on their effectiveness in specific contexts.

This research shows the gap between practitioner expectations of formal education for SE can be reduced through fine-grained alignment of the learning environment with the characteristics of the discipline. While technical knowledge...
Constructive Alignment in SE Education

acquired by students is important in that it acts as a ‘filter’ for graduate employment, of greater impact on the professional competence is the focus on soft and metacognitive skills. These are learnable within a formal education environment, albeit through the application of non-traditional learning models. The final model developed and applied in the research project, Studio Learning, appeared to be effective in addressing issues raised in studies of discipline practitioners and the education literature. The application of Studio Learning within the Murdoch SE programme is discussed in greater detail elsewhere (Armarego, 2007a).

The results of the alignment of this model with the discipline/educational issues highlighted earlier in this chapter can be summarised as the need to:

• Provide students with authentic experiences which address competencies additional to specific discipline knowledge: students were exposed to learning both as a ‘generic’ metacognitive activity, and as a skill to be continually adapted and utilised within a discipline context. Flexibility in thinking - addressing creativity, opportunism and divergency/convergency - was made explicit and strategies to exploit it developed.

• Provide learners with a deep understanding of self and others in complex human activity systems in a collaborative environment: students became aware of and learnt to utilise each other’s strengths and weaknesses in achieving the learning outcomes. They learnt how to ‘jell’, what to do if they did not, and to be empathetic to the contexts of other students. They learnt to value and exploit alternate perspectives brought to a problem by different stakeholders (client, teacher/consultant, other team members) to enrich their learning. They became aware of the need to be self-motivated and learn independently - students were confident in questioning their own and others’ assumptions within the learning environment.

• Allow time to explore new ideas and to reflect on possible processes and outcomes: students were willing to ‘trust’ each other’s knowledge (implicit or not, technical or not), accepting the multi-disciplinary nature of the skills and knowledge required to achieve the learning objectives.

• Be challenged: students were motivated by the (increasing) complexity of the tasks assigned, and were able to focus on cognitive and interpersonal skills to adapt to the changes imposed.

Techniques applied included: providing students with information about learning theory (PBL, situated learning, life-long learning), ensuring ‘higher order learning’ was addressed with greater emphasis on analysis and synthesis rather than application of knowledge within courses, embedding reflective practices within each course (eg journals, performance and team-work reviews), emphasising alternative approaches to problems and ‘rewarding’ diversity of (feasible) solutions, embedding change in all aspects of the problems tackled (changing requirements, scenarios, deliverables, team composition, client contact, etc) to highlight the importance of opportunism, flexibility and adaptability (Armarego, 2007c).

Not only was student feedback positive, and a significant improvement in their assessment marks discernable, but observation and analysis of subsequent learning (Armarego, 2007c) showed strong indications of willingness to transfer knowledge gained, to take control of their learning, and indicated motivation to deeper learning, as indicated by the work of Entwistle and Ramsden (1983).

However, what both practitioner studies (especially the work of Minor (2004)) and this research hint at is the importance of individual characteristics and abilities. Minor’s participants indicated a Personality component to competent practice. Examination of student reflective com-
ments, in conjunction with data regarding student learning, adds another dimension to the issue of education for competent practice.

This research suggests that an alignment between the learner and the (discipline-aligned) learning model enhances student learning of that discipline. However, further research is required to test these findings in the context of student transition to the workplace: at this time, reporting of graduate success (although very encouraging) is only anecdotal.

**IMPLICATIONS FOR THE FUTURE**

An increasing shortage in IT practitioners both through disengagement with the discipline and decreasing enrolments in tertiary institutions suggest an imperative to address the needs of industry and provide graduates with appropriate competency. The implication for education is that it is no longer adequate for academics to only be discipline experts – knowledge and understanding of the complete learning process is vital in achieving this goal, and implies resources committed to appropriate (educational) training. The implication for the learning environment is that it is no longer appropriate to rely on traditional teaching as the basis for the learning process – these methods do not align well with the requirements of the profession, and inhibit many (actual as well as potential) students from engaging with the discipline. This, too, requires resources to be dedicated to invigorating the learning environments provided. The implication for the students themselves is that dependent learning is contraindicated for success in the IT professions. As learning becomes necessarily life-long, students must embrace the skills and knowledge outside the discipline content (the affective and soft skills) required for successful professional practice. From the educational perspective, these must be made explicit by, for example, moving towards student-centred experiential learning models; by embedding higher order, soft and affective skills into the course; and ensuring – through mapping and constructive alignment - that these are a measurable outcome of the learning process.

**CONCLUSION**

This chapter describes a relationship between the characteristics of the discipline and established models of learning. These characteristics inform the development of a conceptual model for SE education, and a learning model that addresses more explicitly the gaps in formal education identified by practitioners. These gaps may be considered as a lack of alignment between the various elements which contribute to graduate competence as practicing professionals in the discipline.

The concept of alignment is well understood and is backed by a body of research literature: in an educational context constructive alignment (eg between objectives and assessment) is considered ‘best practice’; as practitioner studies highlight, in industry alignment between IT practice and formal education is also considered best practice. However, shortfalls in IT professionals in industry, as well as decreasing enrolments and growing student attrition suggest other alignments; those between the discipline, the organisation and education should also be explored. Yet not much work has been published in this area.

The research that this chapter discusses confirms that there is a relationship between characteristics exhibited by learners and the learning environment provided. Students display aptitudes for specific learning environments; these should therefore exploit student learning characteristics since those whose approaches to learning align with the learning model appear to gain increased benefits. If that environment is also aligned with the characteristics of the discipline, it is suggested that students with specific characteristics, taught in a manner that is appropriate to the discipline, have greater potential to becoming competent
practitioners: a case of the sum of the alignments being greater than its parts.

REFERENCES


Constructive Alignment in SE Education

M. J. Smith (Eds.), Designing and Using Human-Computer Interfaces and Knowledge Based Systems (pp. 727-734). Amsterdam: Elsevier.


LTSN. (2002). Constructive alignment and why it is important to the learner, from http://www.ltsneng.ac.uk/er/theory/constructivealignment.asp


UWA. (1996). *Do male and female students differ in their preferred style of learning?* Perth: Institutional Research Unit, University of Western Australia.


**ENDNOTES**

1 When applied to education, alignment refers to the ongoing process of bringing congruence to the declared, learnt and assessed components to guide instruction design and ultimately, student learning. Authors on curriculum alignment agree content, depth, emphasis and cognitive activity match are required for sound alignment (Rothman, Slattery, Vranek, & Resnick, 2002). In the context of this chapter, alignment transcends the educational environment to include discipline, practitioner and student characteristics.

2 Somekh (1989) defines Action Research as the study of a social situation, involving the participants themselves as researchers, with a view to improving the quality of action within it. This research applies the style described as the ‘Deakin’ (Carr & Kemmis, 1986) approach. This has merit in being adopted for studies in educational contexts (Zuber-Skerritt, 1995)
This implied rearranging political and administrative structures in a rationalist way in order to abandon superstition and injustice.

For software development, Zucconi (1995) suggested the underlying disciplines of central importance are psychology, computer science and discrete mathematics, and suggests an IT professional needs to be well organised, able to work as a member of a multi-disciplinary team, and within the scope of the employer’s policies and procedures and society’s tenets.

In general, students exhibited ‘engineering’ styles. As the work of the Felders and their colleagues (e.g., Felder & Spurlin, 2005; Felder & Brent, 2005; Felder & Silverman, 1988) indicate, engineering students are pragmatists with a tendency to narrow technical interests. Converger characteristic, to seek “single, correct answers or solutions to a question or problem” (Kolb, 1995) becomes the dominant learning style.

Exploring the relevance of each new topic (Diverger); making available basic information and methods associated with the topic (Assimilator); providing opportunities to practice the methods (Converger) and encouraging exploration of the applications (Accommodator) (Felder, 1996).