Chapter 8

Engaging Software Engineering Students with Employability Skills

Jocelyn Armarego
Murdoch University, Australia

ABSTRACT

This chapter explores the findings from an Action Research project that addressed the Professional Capability Framework (Scott & Wilson, 2002), and how aspects of this were embedded in an undergraduate Engineering (Software) degree. Longitudinal data identified the challenges both staff and students engaged with. The interventions that were developed to address these are described and discussed. The results of the project show that making soft skills attainment explicit as part of the learning objectives went a long way in assisting students to engage with the activities that exercised these skills.

INTRODUCTION

A number of recent studies have discussed the challenges facing 21st century graduates (e.g. Andrews & Higson, 2008; P. Brown & Hesketh, 2004; Cassidy, 2006). The fast pace of change, affecting the whole of society, has led to a knowledge-driven economy that asks its workforce to “flexibly adapt to a job market that places increasing expectation and demands on them” (Tomlinson, 2012). Being an employable graduate is no longer an automatic result of successful completion of formal study – tacitly acknowledged in the focus on identifying and developing graduate attributes in universities globally as a response to employer concerns.

However, an example definition of graduate attributes: “the qualities, skills and understandings that a university community expects its students to develop during their time at the institution and
consequently, shape the contribution they are able to make to their profession and as a citizen” (Bowden, Hart, King, Trigwell, & Watts, 2000) indicates that, although highlighting the required cognition and character development of students, graduate attributes do not have the focus expected by employers on employability skills. These have been identified as key transferable soft skills and competencies integral to graduate professional practice (Andrews & Higson, 2008). Employability skills are not job specific and “cut horizontally across all industries and vertically across all jobs from entry level to chief executive officer” (Sherer & Eadie, 1987).

As a result of trends in the sector, the nature of ICT employment has evolved and employability has become a major issue (Scholarios, van der Schoot, & van der Heijden, 2004). In the Australian context, a comprehensive study of employability skills conducted on behalf of the federal Department of Education highlighted the importance of abilities sought by employers in addition to technical knowledge and skills (ACCI, 2002). Studies undertaken since 2000 confirm earlier work (Doke & Williams, 1999; Lee, 1999; Snoke & Underwood, 1999; Turley, 1991) that, although the technical competency of ICT graduates can, in general, be assumed, other, softer, skills are considered by practitioners as lacking. In particular the demand and expectation of industry employing ICT graduates has changed: surveys show that they are dissatisfied with graduates in a number of areas (Hagan, 2004).

However, embedding soft skills into undergraduate programmes is not an easy task:

- From the academic’s perspective, this is a tacit aspect of competency and therefore less easy to define and assess – for example, what soft skills should be included? Anecdotal evidence also suggests many academics with high technical expertise find themselves less comfortable in this space, and therefore show a preference for students taking an externally offered course (and therefore without discipline context (Smith, Belanger, Lewis, & Honaker, 2007)) rather than attempting to incorporate these within the discipline
- From the student perspective, technical skills have been viewed as the prime competency to be attained from their studies. Affective skills, cognitive skills, understanding of the ‘context’ in which the task being addressed, are undervalued, often until well after graduation (Lethbridge, 2000).

This chapter explores the findings from an Action Research project that identified one framework where soft skills appropriate to an engineering context were explicitly described - the Professional Capability Framework (Scott & Wilson, 2002) - and how aspects of this were embedded in an undergraduate Engineering (Software) degree. The interventions that were developed to address these are described and discussed. The results of the project showed that making soft skills attainment explicit as part of the learning objectives went a long way in assisting students to engage with the activities that exercised these skills. In effect, soft skills can be mastered in a learning environment that applies active and collaborative learning, acknowledges differences in learning styles, and contextualises the learning of engineering as a profession.

BACKGROUND

Employability

The concept of employability has emerged as a dominant issue in a number of contexts over the last decade (McQuaid & Lindsay, 2005). The quality of the graduate labour market, and graduate ability to meet employer needs, have been the focus of debate across many disciplines acknowledged as
Engaging Software Engineering Students with Employability Skills

impacting on a knowledge-based society. Despite a focus on generic attributes, university graduates lack the ability to apply these in the work place (Precision Consultant, 2007). As a result, there is a widespread perception that employability skills are not effectively developed in an undergraduate programme.

Andrews and Higson (2008) indicate that serious concerns have been expressed about an increasingly wide ‘gap’ between the skills and capabilities of graduates, and the requirements and demands of the work environment they enter. Although expectations of graduates are similar across the globe (that graduates would be employment-ready; equipped with the necessary skills and competencies; and able to work with the minimum of supervision (Harvey & Bowers Brown, 2004)) and, in fact across time, with only the priority changing (DEEWR, 2012), studies show that graduates focus on technical or ‘hard’ skills, despite, since the 1990s, these having being considered less important by employers (Hagan, 2004; Turner & Lowry, 2003).

From the work of McLarty (1998); Tucker, Sojka, Barone, and McCarthy (2000) and others, Andrews and Higson (2008) identify: professionalism; reliability; the ability to cope with uncertainty; the ability to work under pressure; the ability to plan and think strategically; the capability to communicate and interact with others, either in teams or through networking; good written and verbal communication skills; ICT skills; creativity and self-confidence; good self-management and time-management skills; a willingness to learn and accept responsibility, as ideal generic skills and competencies required of graduates in the workplace. Although their work addresses graduates within the business discipline, other studies indicate similar requirements:

- More generally as higher order skills that assist individuals to be more flexible, adaptable, creative, innovative and productive. This requires skills and knowledge to work effectively as a member of a team, cooperate in ambiguous environments, solve problems, deal with non-routine processes, handle decisions and responsibilities, communicate effectively, and see ‘the bigger picture’ in which they are working. It also requires the skills and knowledge that allow individuals to understand the context in which they are working and to apply their existing skills in the new context (Curtin (2004) cited in DEEWR (2012))

- In specific disciplines, such as engineering and ICT, the focus of this chapter:
  - Professional skills for ABET accreditation for both engineering and computing programmes in the USA (namely: an ability to function in teams; an understanding of professional and ethical issues and responsibility; an ability to communicate effectively with a range of audiences; the broad education necessary to understand and analyse the impact of solutions in different contexts; a recognition of the need for, and an ability to engage in lifelong learning and continuing professional development and a knowledge of contemporary issues (ABET, 2012a, 2012b))
  - For Engineers Australia as professional and personal attributes (namely: ethical conduct and professional accountability; effective oral and written communication in professional and lay domains; creative, innovative and pro-active demeanour; professional use and management of information; orderly management of self, and professional conduct; effective team membership and team leadership (EA, 2011)).
Studies in the disciplines of ICT indicate technical abilities are covered in curricula, and that students and employers are well informed as to the nature of these (Lowry & Turner, 2005). However, a number of practitioner studies attest to a lag in other abilities. Examples across different ICT sub-disciplines include international studies (Lee, 1999, 2004; Lethbridge, 2000; Noll & Wilikens, 2002; Turley & Bieman, 1995; Zwieg et al., 2006) as well as studies in an Australian context (Kennan, Willard, Cecez-Kecmanovic, & Wilson, 2009; Scott & Wilson, 2002; Scott & Yates, 2002; Snoke & Underwood, 1999; Turner & Lowry, 2003)) and highlight the need for:

- An understanding of business functions and organisational knowledge
- The ability to teach themselves what they need to know to perform the task successfully
- Interpersonal skills and personal attributes;
- The adaptability and flexibility required to be ICT practitioners
- Generic attributes such as lifelong learning;
- As well as career resilience (Waterman, Waterman, & Collard, 1994).

Few studies address the skills and knowledge needed in Software Engineering specifically. Turley and Bieman (1995) examined professional Software Engineers in an attempt to identify the competencies and demographics that contribute to ‘excellence’ in performance. They identify four categories of competencies which are statistically significant in differentiating between exceptional (XP) and non-exceptional (NXP) performers. Interpersonal Skills (helping others); Personal Attributes (proactive role with management, exhibiting and articulating strong beliefs and convictions and maintaining ‘big picture’ view); Task Accomplishment (mastery of skills and techniques) were the top competencies displayed by XP performers. The final category - Situational Skills (e.g. responding to schedule pressure by sacrificing parts of design) - was significant as a competency not exhibited by XPs. Lethbridge (1999)’s study highlighted the gap between formal learning and practitioners’ perception of importance. Of the top ten gaps, soft skills figure at position 1 (negotiation – 84% gap between learning and importance in practice; position 3 (leadership – 73%); and position 7 (ethics and professionalism – 63%).

Australia, similar to many OECD countries, has a great deal of emphasis on the development of knowledge, skills and attitudes of its citizens, representing ‘human capital’ particularly in a knowledge-economy. This requires an understanding of the relevant skills and systematic development of these through effective education and training (OECD, 1996), as well as a commitment to lifelong learning. The comprehensive study of employability skills conducted on behalf of the Department of Education defined employability skills as “skills required not only to gain employment, but also to progress within an enterprise so as to achieve one’s potential and contribute successfully to enterprise strategic directions” (ACCI, 2002, p. [3]). This definition highlights the importance of abilities sought by employers in addition to technical knowledge and skills.

Scott and Yates (2002) and Scott and Wilson (2002) undertook Australian studies tracking the experience of successful recent graduates from all 38 publicly funded Australian universities. Their work parallels an extensive government-initiated project analysing responses from recent graduates to the CEQ (Course Experience Questionnaire). Scott, Wilson and Yates operated from the assumption that graduates working in professional practice for between two and six years are well positioned to identify what is likely to be most relevant for those currently studying at university. Such people have sufficient experience to know what counts in the real world of the profession.
whilst not being too far away from their university courses to have forgotten what was covered.

In both the studies Emotional Intelligence ranked highest in importance, dominating the factors identified by graduates as important to their professional careers, closely followed by Intellectual Capability, addressing generic issues such as abstraction and contingency, while profession-specific knowledge ranks relatively low. The ability to work in teams, particularly cross-disciplinary teams that are common in the ICT workplace, is also considered vital.

Support for the results of global and Australian studies is provided by the surveys of ICT employers and ICT graduates in the workplace carried out in the ICT-DBI study (Koppi & Naghdy, 2009), commissioned by the Australian Learning and Teaching Council. The survey of ICT employers in this study reported that recent ICT graduate recruits met about half of the knowledge and skills required (but with the caveat that these were not at the bleeding edge of technological innovation employers and graduates wished for). However, the graduates only met some of employer needs concerned with understanding business processes, project management knowledge, and written communication ability. More than half of the employers indicated that graduates did not meet their needs for commercial awareness, and the interpersonal skills provided in the survey met less than half of the needs of employers. These findings indicate that the university ICT curriculum needs to be brought into line with industry requirements with respect to the development of employability-relevant soft skills.

So, the work of Litecky, Arnett, and Prabhakar (2004) may be a valid summary of the place of soft/professional/employability skills. They suggest technical skills are used a ‘filtration’ in the hiring process – technical screening by an employer is used to eliminate some candidates from the pool of potential candidates and to pass others on to the second stage. They argue that this second stage can reasonably be expected to consist of information on a different set of skills – this stage is based on the perceived soft skills of the candidate, obtained through face-to-face assessments. This second set of skill data is the more important for the final hiring of the candidate.

Thus, while the successful professional must possess a high level of profession-specific technical expertise, such skills have little value without the ability to handle uncertainty, deploy appropriate components of one’s repertoire of generic and profession-specific expertise, accurately diagnose the unexpected and work productively with people from a wide variety of backgrounds.

**Frameworks for Professional Competency**

Scott and Yates (2002) and Scott and Wilson (2002) discuss the findings of their studies in relation to a Professional Capability Framework that identifies competency in five dimensions consisting of: Emotional Intelligence – personal and interpersonal; Intellectual Capability; Professional specific skills and knowledge; and Generic skills and knowledge. An Educational Quality scale addresses issues of appropriateness, authenticity of tasks and assessment (Scott & Yates, 2002).

The Professional Capability Framework is further refined (Scott & Wilson, 2002): Emotional Intelligence (personal and interpersonal (now social)) becomes Stance and Intellectual Capability is now defined by two components: Way of Thinking (incorporating cognitive intelligence and creativity) and Diagnostic Maps (developed through reflection on experience). The characteristics of the revised framework are taken from Scott and Wilson (2002, pp. 7-9):

A. **Stance**: Emotional Intelligence-social includes an ability to empathise with the perspectives of others, to work constructively in a team, engage in reciprocal relationships, to be patient and to allow others ‘room’ to do things for themselves. Emotional Intelligence- personal includes a capacity and willingness to try new things, take informed
Engaging Software Engineering Students with Employability Skills

In summary, the framework supports their findings that, while technical and generic skills are necessary (D and E), they are not sufficient for competent professional practice: a high level of social and personal emotional intelligence (A); a contingent way of thinking (B); a set of ‘diagnostic maps’ (C) developed from handling previous practice problems in the unique work context (Scott & Wilson, 2002) are imperative. The framework has been applied or adapted to a number of disciplines, including higher education.

The Australian study of employability skills noted above (ACCI, 2002) proposed an Employability Skills Framework, originally published in 2006 (DEST, 2006) and in the process of being updated. This later version emphasises the “underpinning skills and knowledge that enable individuals to perform effectively in employment contexts and to apply their technical or discipline specific skills in different contexts” (DEEWR, 2012 p 6). The report highlights the fact that, while consensus exists on the types of skills sought, employers are more interested in the outcomes - that is, individuals demonstrating these skills, and the attitudes that are outcomes of proficiency in these skills. As one example provided: “adaptability is the result of an ability to take risks and respond to challenges and to adapt and apply prior learning” (DEEWR, 2012). The Employability Skills Framework will encompass both employability skills and aspects of the context which impact upon an individual’s ability to develop and demonstrate these skills. Technical or discipline specific skills are detailed in educational curricula, while the core language, literacy and numeracy (LLN) skills of reading, writing, oral communication, numeracy and learning are addressed in the Australian Core Skills Framework (McLean, Perkins, Tout, Brewer, & Wyse, 2012).

risks, tolerate uncertainty, ambiguity and change, admit and learn from errors, defer judgement, pursue excellence and persevere, to work independently, withstand personal attacks, behave ethically, keep work in perspective, lead a balanced life and to ‘pitch in’ and do ‘drudge work’

B. Way-of-thinking: Reflects an ability to think ‘contingently’ and creatively, to be able to trace out and assess the consequences of different options for action, to identify and accommodate conflicting interests and perspectives, to set priorities, identify the core issue in any situation and to think holistically, laterally and iteratively not rigidly, technically or in a linear manner. This way of thinking cannot operate well if the individual lacks the emotional intelligence to work with continuing complexity, uncertainty and ambiguity and in collaboration with a wide range of people

C. Diagnostic Maps: Refer to that which gives meaning to what an outsider or novice would find a complex and hard-to-fathom in a set of work-specific factors and relationships. These are generated through challenging practice, reflection, and transfer. These ‘maps’ are most effectively developed if the individual has been able to deploy a high level of social emotional intelligence to build networks

D/E. Generic Skills and Technical Expertise: Such knowledge actively assists with the process of problem interpretation and diagnosis. Generic skills such as the ability to effectively undertake team work, network, communicate and present ideas, manage project, learn one’s organisation, mentor, carry out self-managed learning or to organise one’s work and life, also have importance.
Engaging Software Engineering Students with Employability Skills

Soft Skills in the ICT Curriculum

Bentley, Lowry, and Sandy (1999) suggest a developmental process in which personal attributes, which influence intellectual abilities and skills, are applied to the acquisition of knowledge in order to enable the development of higher cognitive activities. They note that, at the end of the educational process, students must be able to apply knowledge to new situations and problems. This ability to transfer requires certain generic intellectual abilities and skills, which, they suggest, although highly valued by employers of graduates, are sometimes given only ‘lip service’ in tertiary education curricula. The personal attributes identified as important in their model include attributes like curiosity, risk taking, personal discipline and persistence, which can influence in important ways the successful application of intellectual skills and abilities to knowledge and hence support the higher orders of thinking. Earlier Turley (1991) had suggested a significant area for research was to explore how competencies are reinforced. He concluded that education needed to support the development of differential skills (namely interpersonal skills and personal attributes) through the creation of learning situations which stress these.

The approach taken for examining education for ICT specialisations in general (and hence to some extent taking up the challenge in the discipline) has been the revision of the various model curricula (ACM/IEEE, 2008; GSweE, 2009; LeBlanc & Sobel, 2004; Topi et al., 2010) and the Bodies of Knowledge which underpin them (e.g. SWEBoK (Bourque & Dupuis, 2004; Fairly, 2013)) to identify guidelines that support soft skills learning. So, for example a curriculum guideline may state:

**Guideline – Students should be trained in certain personal skills that transcend the subject matter:** The skills below tend to be required for almost all activities that students will encounter in the workforce. These skills must be acquired primarily through practice:

- **Exercising critical judgment:** Making a judgment among competing solutions is a key part of what it means to be an engineer. Curriculum design and delivery should therefore help students build the knowledge, analysis skills, and methods they need to make sound judgments. Of particular importance is a willingness to think critically. Students should also be taught to judge the reliability of various sources of information.
- **Evaluating and challenging received wisdom:** Students should be trained to not immediately accept everything they are taught or read. They should also gain an understanding of the limitations of current SE knowledge, and how SE knowledge seems to be developing.
- **Recognizing their own limitations:** Students should be taught that professionals consult other professionals and that there is great strength in teamwork.
- **Communicating effectively:** Students should learn to communicate well in all contexts: in writing, when giving presentations, when demonstrating (their own or others) software, and when conducting discussions with others. Students should also build listening, cooperation, and negotiation skills. (LeBlanc & Sobel, 2004, p. 40)

How these and similar skills should be acquired and honed within the learning environment is less tangible (Voogt, Erstad, Dede, & Mishra, 2013).

In general, learning in higher education is complex (Claxton, 1998). In order to achieve complex learning, appropriate environments and opportunities should stimulate a learning which is a “constructive, cumulative, self-regulated, goal oriented, situated, collaborative and individually different process of knowledge building and meaning construction” (De Corte, 2000, p. 254). However, this raises an additional issue: Benson (2003) notes that within the emerging ICT discipline of the 1970s, academics were migrants to
Engaging Software Engineering Students with Employability Skills

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, with many migrating from engineering and manufacturing. Therefore the degree of alignment (or dissonance) between practice and education, in effect between the theory-in-practice (what practitioners do and what competencies they need to do what they do) and the espoused theory (what formal education says practitioners do, and how students are taught to do it) (Argyris & Schö, 1974) may be exacerbated by the learning/teaching models academics bring to the learning environment, usually heavily influenced by their own discipline learning.

Matching the gaps identified by practitioners, and learning models that purport to focus on these, shows that non-traditional approaches provide leverage for a graduate entering the profession of Software Engineering. Such approaches are based on active learning models, addressing learning as more discursive and collaborative. Active learning approaches are seen to create situations which engage students in such higher order thinking tasks as analysis, synthesis and evaluation (Bloom, 1956). By contrast, non-active learning are directed to absorption and imprinting. Active learning methods also attempt to develop the cognitive (knowledge, understanding and thinking) and affective (emotive) dimensions of the learning process in such a way that learners’ active involvement in the learning is improved (LTSN, 2002).

In developing a framework for advanced active learning Horvath, Wiersma, Duho, and Stroud (2004) propose a topography based on the focus of the approaches and the nature of the applied methods (see Figure 1). These are described as a continuum from instructive through explorative.

Figure 1. Topography of advanced approaches to active learning (adapted from Horvath et al. (2004))
Engaging Software Engineering Students with Employability Skills

to constructive, with the latter assuming that learners construct knowledge for themselves by creative activities such as planning and design. The orthogonal axis describes the focus of the activities: from a priority given to sensory-motor and conceptual activities of individual students to the development of knowledge by a group or community of learners.

However, learning approaches that involve critical thinking and reflection are a challenge to students: in ICT they expect to be taught formulaic and recognised methods that will allow them to build successful systems. Banks (2003) even suggests an approach that requires reflective and active questioning that challenges previously held beliefs may be inappropriate for undergraduate students.

Development of Competency

McCracken (1997) suggests that formal education provides the entree into the discipline. A principled academic education enables the graduate to become, over a period of years after graduation, a member of the profession. McConnell and Tripp (1999) go so far as to suggest at least four years of apprenticeship necessary for Software Engineers.

Work on the transition from study to workplace (e.g. Lee (2004)) has shown that the onus of teaching themselves what they needed to know in order to perform the task successfully is one of the ‘reality shock’ involved in the socialisation of new graduates to work. 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). Full integration into the organisation was seen to take up to two years, during which time they were not considered an ‘insider’ and ‘working professional’. What the practitioner studies indicate is that this is too long (Benamati, Zafer, Ozdemir & Smith, 2010), and assumes too much ‘on the job’ training. The reality is that new graduates are expected to perform on the same level as their experienced counterparts - the best that can be hoped for is a sympathetic, experienced mentor as coach. Higher education is required to reduce the amount of time taken to become a competent member of the profession – in effect being more than novices on graduation.

Within education competence or mastery are often linked to the concept of learning objectives or outcomes, and usually are seen as positions on a continuum which spans pre-novice to expert. Jonassen and Grabowski (1993) describe such a continuum as leading from ignorance to expertise. Dreyfus and Dreyfus (1986) and later Dreyfus (2001) developed a richer skills acquisition model that focusses on adult education. This model is based on five (then six) stages: novices and advanced beginners apply rules and maxims to solve problems; competence is process driven and problem-solving conscious; proficiency applies pattern recognition arising from extensive experience to identify the problem as intuitive reaction; intuitive situational response is automatic in the expert, based on abstract representations formed through reflection on experience; mastery provides a sense that there is no one right thing to do and that improving is always possible. Masters respond immediately to the whole meaningful context: reaching a new level of skilful coping beyond expertise and developing a ‘style’. In relation to education, the Dreyfuses also suggest that the higher modes of functioning – intuitive expertise and mastery – require risk taking (and hence emotional engagement), direct experience and active involvement in the company of experts.

An additional concept is important here - competence is defined, contained and developed within communities of practice (Wenger, 2000). Since the discourse of a discipline is central to the way its knowledge is constructed and transmitted, the means by which learners acquire discursive knowledge is important. While expert members of a discipline share knowledge about discursive practices in their community, this is mostly tacit, and traditionally given little emphasis within the (formal) educational environment.
Therefore there is marked difference between the fragmentary knowledge structures of novice understanding and the integrated knowledge structures which underpin more robust knowledge and flexibility exhibited by the expert knower (Wood, 1999). The theories of learning highlight aspects of knowledge and skill development that are pertinent: advanced competence in a discipline is based on a rich framework of understanding and a commensurate reduction of effort in further learning achieved through:

- Problem-solving practice
- Correspondence with the learner’s intuitive model of the phenomenon (and all that implies about previous experience, belief systems etc.)
- Ease of transfer (with its focus on strategic thinking and metacognitive skills)
- Facility with multiple representations – translation between different models facilitates the understanding of concepts, whilst the models themselves support differing insights, reasoning and problem-solving occurring within a social and cultural context (and all that implies about dialogue, self-explanation)
- Shared meaning where certain activities are seen as authentic and the discursive practices of a discipline are integrated.

Such characteristics of expertise are viewed as fundamental skills for Software Engineers.

This chapter argues that industry competency expectations of graduates assume cognitive skills related to higher order learning, the development of emotional intelligence, and problem-solving strategies, and therefore map to a higher level of competency on the novice-to-master continuum than is normally assumed for graduates. Practitioners look for graduates who are flexible, adaptable in the organisational environment and can continue learning.

### ADDRESSING PRACTITIONER EXPECTATIONS

The purpose of the research reported in this chapter was to develop and implement learning strategies which address the issue of aligning the competency expectations of SE practitioners with formal education for SE, so that, by addressing the practitioner gaps, graduates gain leverage in becoming competent professional practitioners early in their career.

Constructivist learning, a theoretical framework based on the work of Papert (1980), provides the greatest opportunity to achieve this goal. Constructivism distinguishes itself from more traditional instruction, in part, by the degree of active learner engagement as well as the assumption that learners have the ability to create meaning, understanding, and knowledge - learners develop their own reasoned interpretations of their interactions with the world. Perhaps more important, constructivist learning environments allow learners to share and collaboratively reflect on these cognitive artefacts.

Friedman (2001) indicates that experiences of the pioneers show that problem-based learning, project-based learning and learning by doing are the methods that offer the largest potential for constructive learning, in particular, in groups and communities. The first two approaches can be differentiated by saying that problem-based learning concentrates on better understanding and the solving of recognised problems, while project-based learning focuses on the end product. Hence, project-based learning is more or less an artefact production process, while problem-based learning is a knowledge development process. They can appear in practice side-by-side or interwoven. Learning by doing places the learners in direct contact with the subject matter and facilitates finding information through social communication.

Other research on student learning of the discourse of a discipline draws on the metaphor of apprenticeship, described as some gradually mentored pathway to membership. Learners can
be enculturated in the discipline through both being socialised to the ‘forms of talk’ (Berkenkotter & Huckin, 1995) in the community, and gaining conceptual knowledge. Students are seen to absorb the disciplinary practices rather than learn them through explicit teaching. Lave and Wenger (1991) considered the importance for newcomers of grasping knowledge and skills, achieved by means of ‘legitimate peripheral participation’. In their view, the notion of participation deals with the process of situated learning, an integral and inseparable aspect of social practice which encourages newcomers to become part of a community of practice. Earlier, Lave, Smith, and Butler (1988) had suggested that learning is a process that involves becoming a different person with respect to possibilities for interacting with other people and the environment. The individual is no longer the same individual with new skills, but is a new person who has become more enculturated into the practice, negotiating meanings based on experiences as a student.

Students bring a complex assortment of beliefs, past experiences and expectations to a learning situation, which influence the approach to learning they take. In turn, this affects the quality of their learning outcomes, and reflects on their future learning intentions and behaviours (Prosser & Trigwell, 1999). While the achievement of high quality learning is important in all graduates, it has an extra relevance where the context of practice is continually changing and professions are continuously developing. There is a need, therefore, to identify aspects of the learners’ conceptions of learning (Meyer & Shanahan, 2000) and approaches to it so that appropriate support can be provided. The approach to learning that a student takes is also sensitive to the context in which learning is done, with a demonstrated correlation between more advanced conceptions (e.g. abstraction of meaning and understanding of reality) and a deep approach to learning (see Wilson and Fowler (2005) for a discussion of the relevant studies). This conception to a large extent determines the student’s expectation of what the learning process and teaching entail. Thus, for some students, the ability to take a deep approach (and hence the quality of their learning) appears to be limited by the conception they hold of learning.

**Context for the Project**

Over a number of years, an action research project was undertaken at this university with the objective of graduating software engineers who could engage with practitioner requirements for soft skills. A variety of interventions, based on non-traditional learning environments, were designed and implemented to address these non-technical skills, and both qualitative and quantitative data collected and analysed. A practitioner perspective on how the ‘work’ of the discipline is undertaken also coloured the conceptual framework developed through the interventions.

The curriculum context for this project was an undergraduate 4-year degree in Software Engineering (BE (SE), accredited by both Engineers Australia (EA) and the Australian Computer Society. After a common first year students transitioned into the engineering discipline of their choice. The curriculum framework comprised elements of Computer Science (to cover fundamental aspects to form the basis of technical knowledge and skills in software and hardware), Software Engineering (a focus on SE theory and practice that forms the basis of core knowledge and skill in software development and evolution), and Engineering (knowledge and skills in engineering practice and principles including those elements of EA curriculum requirements not covered in the previous components. These are common to all Engineering students within the School).

It should be noted at this point that all content material was available online, so lectures and tutorials could be replaced by workshop-based discussion and exercises. Briefly, the **Software Factory** presented a coherent system and learning context. Topics were categorised mnemonically
as sections. This allows for ‘chunking big’ and focusses on connections between topics in the same category for content- and context-dependent knowledge construction (Jonassen, 1992). Each course has a comprehensive study programme comprising a set of topics across relevant sections. Once authenticated as a member of the class, a student was able to access the unit material both through the **SENavigator** (see Figure 2) or the Production Line (see Figure 3). Within each course topics are sequenced and displayed on a map that provides alternative routes from commencement to completion. To a certain extent these provide choice in the order of topics studied and allow students to vary the sequencing of content, although support in the form of workshop schedules dictate the dates by which topics must be completed. Nevertheless, students had the flexibility to access any topic to assist them in solving problems encountered throughout the course.

While the learning environment has as its focus activities/real-world problem solving, on-line/interactive activities cease to be meaningful if the student hits a snag and is unable to progress from there. The purpose of scaffolding is to provide activity-sensitive help mechanisms. Examples include exercises; explanations and background information; monitoring tools (to help students keep track of their progress); modelling scratchpads (software tools to create and manipulate models); problem hints; process coordinators (which guide the students through the complete task cycle); and planning tools (Jong, 2006).

The **Software Factory** provided examples of all these scaffolds, as appropriate, both as purpose-built activity help and underlying manuals. The former was accessible through a ‘help’ icon on an activity screen, while the latter is best demonstrated through the underlying help in the FM (Formal Methods) topics, where help is activated through ‘hot’ spots in the notation itself. Both of these mechanisms are not imposed on the student, but are readily available. Links to the help are

---

**Figure 2. SE navigator for RE course**

![SE navigator for RE course](image)

---

<table>
<thead>
<tr>
<th>Section II - REQUIREMENTS MODELLING - Defining Requirements Engineering and the need for RE modelling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RM 000</strong></td>
</tr>
<tr>
<td><strong>RM 010</strong></td>
</tr>
<tr>
<td><strong>RM 020</strong></td>
</tr>
<tr>
<td><strong>RM 030</strong></td>
</tr>
<tr>
<td><strong>RM 100</strong></td>
</tr>
<tr>
<td><strong>RM 110</strong></td>
</tr>
<tr>
<td><strong>RM 120</strong></td>
</tr>
</tbody>
</table>
Engaging Software Engineering Students with Employability Skills

Figure 3. Production line for RE course

Figure 3. Production line for RE course

seamless, which enables the student to maintain focus on the learning activity, rather than on the task of retrieving aid.

Interventions 1 and 2 were situated within the first of the eight core SE courses (four courses is a full semester load). As noted above, all engineering students had undertaken a common first year: they had generally been immersed in a scientific/engineering paradigm where problem-solving through laboratory procedure, repeatability of experimentation and rigour in mathematics were key learning objectives. The result of this was strong preconceptions about both teaching and learning. Requirements Engineering (RE), offered in semester 1 of the second year of study addressed RE in an object-oriented environment. Students were exposed to O-O principles, tools (Rational Rose at that time) and techniques based on UML in order to construct a Requirements Specification for a given problem. The course was taught in workshop mode (two sessions of 2-hours twice per week).

By the time Intervention 3 was to be operationalised, the curriculum had been modified from the eight SE courses to three Design Studios in each specialisation. Two Studios comprised a full semester load, with 20 hours ‘student workload’ allocated to each. Of this time, up to ten hours could be class contact. The specialisations would only kick in in year 3 (giving two common engineering years instead of the previous one). The original RE course became the foundational content of the first SE Studio with additional material absorbed from another original SE course.
INTERVENTION OVERVIEW

The three cycles of intervention of this project explored alternate learning models to evaluate their appropriateness for addressing a shift in focus from technical competency to the soft and metacognitive skills that enable the competent practice of SE. These were based on *Cognitive Apprenticeship*, as developed by Collins, Brown, and Newman (1989); *Problem-based Learning* (PBL), which draws from several theoretical traditions including pragmatism (Dewey, 1916), cognitive psychology (Piaget, 1968) and social constructivism (Vygotsky, 1978), and a modified *StudioLearning* approach that incorporates elements of PLB into a reflective practice model. Kuhn (2001) describes the characteristics of the Studio model: open-ended problems are solved, using a variety of media, through rapid iteration of design solutions, with frequent critique and contingent decision-making. To re-iterate from definitions quoted previously in this chapter, graduates from these learning environments should: see ‘the bigger picture’ in which they are working; be more flexible, adaptable, creative, innovative; cope with uncertainty; cooperate in ambiguous environments; deal with non-routine processes; plan and think strategically and be able to transfer; handle decisions and responsibilities; acquire good self- and time-management skills and self-confidence as professional software engineers (Andrews & Higson, 2008; Curtin, 2004). The overarching aim was to expose students to a learning environment that would address components of the Professional Capability Framework as appropriate to achieve these outcomes. A mapping provides an indication of the synergies between them. However, it should be noted that dependencies that exist between soft skills (see the description of the components of the Professional Capability Framework for an example of this) meant that students were exposed to soft skills other than those explicitly noted in Table 1. As one example ‘cooperate in ambiguous environments’ assumes teamwork, communications, and management (self and task) skills. Each intervention strategy addressed specific concerns and, through evaluation of and reflection on each, strategies are refined for subsequent iterations.

**Intervention 1: Cognitive Apprenticeship**

The first intervention focussed on authenticity and transfer of skills acquired to other courses and, eventually, to the profession. The authenticity referred to providing students with an environment where they were treated (and acknowledged) as novice software engineers rather than generic engineering students. The aim of this strategy was to engender a strong sense of community. Transfer (the skilled application of understanding) has been acknowledged as a significant component of

<table>
<thead>
<tr>
<th>Framework Dimension</th>
<th>Soft Competency</th>
<th>Intervention</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stance</strong></td>
<td>cope with uncertainty</td>
<td>CreativePBL</td>
</tr>
<tr>
<td></td>
<td>cooperate in ambiguous environments</td>
<td>StudioLearning</td>
</tr>
<tr>
<td></td>
<td>handle decisions and responsibilities</td>
<td></td>
</tr>
<tr>
<td><strong>Way-of-thinking</strong></td>
<td>see ‘the bigger picture’ in which they are working</td>
<td>Cognitive Apprenticeship</td>
</tr>
<tr>
<td></td>
<td>be more flexible, adaptable, creative, innovative</td>
<td>CreativePBL</td>
</tr>
<tr>
<td></td>
<td>plan and think strategically</td>
<td>StudioLearning</td>
</tr>
<tr>
<td><strong>Diagnostic Maps</strong></td>
<td>deal with non-routine processes</td>
<td>Studio Learning</td>
</tr>
<tr>
<td><strong>Generic Skills and Technical Expertise</strong></td>
<td>acquire good self- and time-management skills</td>
<td>Cognitive Apprenticeship</td>
</tr>
<tr>
<td></td>
<td>acquire self-confidence as professionals</td>
<td>CreativePBL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StudioLearning</td>
</tr>
</tbody>
</table>
problem solving success – which in turn requires self-regulation metacognitive skills (directing, monitoring and evaluating learning) and engagement with multiple perspectives on the problem, its abstraction and its solution. Transfer is seen to be enhanced within a socio-cultural context (Brown, Collins, & Duguid, 1989) (in this case the community mentioned here) where the skill is de-coupled from the task (abstracted) and the problem-solving interpreted through mental modelling (Gott, Hall, Pokorny, Dibble, & Glaser, 1993; Patry, 1998). Peer-to-peer knowledge transfer is also required (Budgen, 2003), achievable through collaborative work.

In Cognitive Apprenticeship settings, learning is considered a process of active knowledge construction that is dependent on the activity, discourse, and social negotiations that are embedded within a particular community of practice (Collins et al., 1989). The curriculum for RE was addressed as a two-cycle spiral: the workshops during first part of the semester (8-9 weeks) focused on learning and practicing use of the tools, gaining an understanding of the conceptual framework (in this case object-orientation principles), and an appreciation for the context in which professionals practice (e.g., historical overview; issues in RE theory and practice; organisational involvement; group dynamics). The second part of the semester focused on issues of team work and knowledge transfer - students were involved in a group project that required them to apply the tools to model a complex problem: the development of a ‘complete’ Requirements Specification. In broad terms, the phases (see Table 2) of the Cognitive Apprenticeship model were traversed throughout the semester, though without a clean break—the focus of the class sessions changed, but the ability to revisit any phase as required existed. The value of this model is its alignment with practitioner perception of learning in the discipline: they indicated an apprenticeship model of on-the-job learning for graduates was appropriate.

As applied in the SE course, the teacher initially modelled effective practices (e.g., given an informal system description, how do we create appropriate UML diagrams; which are appropriate; how are different elements of the notation applied, and when and why; etc.), demonstrating tool use, techniques and applying notation to address the problems presented. The students then undertook tasks they would encounter as practicing professionals, requiring proficiency with notations and tools, but also an appreciation of the context in which these must be applied (e.g., how/why different types of systems need different types of modelling – not all systems are necessarily best modelled as o-o, etc). This required an understanding of the underlying conceptual frameworks and paradigms used in the domain. Because these skills were all new to them, the students were closely coached by the teacher, both

<table>
<thead>
<tr>
<th>Phase</th>
<th>Component</th>
<th>Class Session#</th>
<th>Activities &amp; Teacher Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Modelling</td>
<td>1-6</td>
<td>Demonstration of the task as a process. Example approaches and sample solutions are provided as a basis for comparison and critique. Teacher explains strategies applied and use of modelling tools (e.g. notation) explicitly.</td>
</tr>
<tr>
<td>II</td>
<td>Coaching</td>
<td>7-16</td>
<td>Critique and whole class discussion of individual approaches applied. Focus is on exploration of multiple perspectives and the reasoning process.</td>
</tr>
<tr>
<td>III</td>
<td>Scaffolding</td>
<td>17-20</td>
<td>Teacher’s role is to question, prompt and encourage students to stay on task.</td>
</tr>
<tr>
<td>IV</td>
<td>Fading</td>
<td>21-26</td>
<td>Student collaboration and peer discussion lead to a negotiated solution for submission.</td>
</tr>
</tbody>
</table>

#The course comprised of 2 * 2 hour workshop sessions for 13 weeks
Engaging Software Engineering Students with Employability Skills

individually and in their groups, to apply a process for modelling each task as they reason about the issues being raised. At this point the learning becomes student-driven (though not yet student-centred). Students practice the techniques previously modelled by the teacher, with the step-by-step process acting as scaffolding. Coaching took the form of side-by-side work, protocol analysis (what is the goal and what steps are needed to get there) and reflective discussion of how the problem could be tackled and why specific models were a ‘better’ representation of the problem than others. Whenever the students reached an impasse, and were unable to continue or complete the task independently or with assistance from group peers, the teacher once again modelled the appropriate approach in a protocol analysis environment, for all students.

Gradually, students were able to complete tasks more independently, with the final class assessment item requiring the development of a complete model of a problem, including critique and justification of the approach taken, with minimal support from the teacher. Other assessment items (concept maps, portfolio) addressed the level of conceptual understanding, and student’s willingness to explore outside the boundaries provided within the course.

The learning environment was one that some students found difficult to assimilate. Across a semester, ambivalence was quite noticeable: while some students appreciated it as co-operative and interactive (example student feedback included: class discussions are very useful; casual workshop environment), others felt it shifted the burden too heavily to their shoulders (far too much content to read; too much workload leading to no marks). Although collaboration strategies had been made explicit (with team roles and duties identified), in practice students were also unhappy about the amount of collaboration required: the need to negotiate with team members, explore alternatives and resolve conflict, co-ordinate tasks and produce a deliverable that had a unified look-and-feel, taxed some students (this leaves little time for other subjects; not clearly stated what we are required to do in terms of assessed work) while others could relate the experience to the workplace (can see industry advantages; shows how to analyse stuff). In almost every case of a request for an extension beyond the due date, the comment was on the lines of the work is done, just not put together.

In effect, students saw software development as fundamentally scientific (where following a defined process will always lead to a quality product (Pfleeger, 1999)) and, although they accepted the collaborative nature of the process, were not comfortable with both the breadth of acceptable solutions to a problem and the amount of group work expected. This perspective is not unexpected – as Baxter-Magnola (2001) suggests, students at low intellectual (or epistemological) stages of development either believe that every intellectual and moral question has one correct answer and their (competent) teachers know what it is or are transitioning to believing that some knowledge is certain, and that making judgements following logical procedures prescribed by authority deserves full credit. Challenges to their belief systems within the courses they take and interactions with peers are necessary for them to gradually come to believe in the validity of multiple viewpoints. However, discussions during the class suggested these perceptions were very little changed.

Evaluation of the Cognitive Apprenticeship model in terms of the Professional Capability Framework indicated low levels of both component A and D. In relation to Emotional Intelligence - social, although students were required to work collaboratively, there was little demonstration of “ability to empathise with the perspectives of others, engage in reciprocal relationships, to be patient and to allow others ‘room’ to do things for themselves” (Scott & Wilson, 2002, p. 7). In terms of Emotional Intelligence - personal, students’ capacity and willingness to try new things (in this case a different approach to learning), toler-
Engaging Software Engineering Students with Employability Skills

ate uncertainty, ambiguity and change were also not well demonstrated. Students exhibited some of the traits of surface learning - they focussed on learning the tools and techniques of SE (component E) at the expense of a more expansive view of the discipline: they did not see themselves as acquiring the more generic skills (component D) valued by practitioners, with the majority of students focussed on being able to apply the tools and techniques in order to achieve a pass. The low level of Stance affected other components as well, specifically components B (Way of Thinking) and D (development of Diagnostic Maps to deal with complexity at a non-novice level).

As a positive note, when the same student cohort undertook the follow-on course they showed an improvement in student success rate (in raw marks) on a similar task (creating a Requirement Specification), achieved closer to schedule. This suggests a measure of transfer (component C) had occurred – students were able to complete the task given in an appropriate time frame and at an appropriate level of competency. Also on the positive side, although attendance at workshops was not compulsory, participation was very high (generally 90 - 100%): active participation on the part of most students is seen as an indicator of engagement (helpful class sessions; lab teaching is good; fun topic, encouraging; good workshops).

In summary, the evaluation suggested the Cognitive Apprenticeship model could be applied reasonably successfully from an academic perspective. However, the majority of students were not comfortable with a ‘master’ who, towards the end of semester ‘faded’: they acknowledged that this placed the onus on them to do the learning (which they did not necessarily accept). A significant finding of this intervention also related to student emphasis on ‘correct’ answers to problem solving undertaken – the master should always be right. The results of this cycle therefore indicated the intervention was only achieving some part of the goal: while students appeared more confident, in subsequent courses, in applying the knowledge they had gained, they still expected to be explicitly taught: that is, the master/apprentice relationship was assumed whether it remained appropriate or not.

In effect, elements of an ‘incorrect’ learning environment were identified in the evaluation of that intervention. Patel, Kinshuk, and Russell (2000) argue that learners in an ill-fitting setting focus on skills that will yield higher grades as an immediate objective. With the relevance of domain knowledge not fully understood, cognitive skills related to exam techniques acquire importance though they do not model real life situations. The learning, in many cases, is reduced to assignment hopping with ‘just-in-time’ and ‘just-enough’ learning to fulfil the assessment tasks. These are also characteristics of surface learning. The Apprenticeship cycle exhibited some of these traits – students focussed on learning the tools and techniques of SE at the expense of a broader (and more abstract) understanding within the discipline.

Apprenticeship models may be seen as limited in an academic context for other reasons which have heightened significance for the ICT discipline:

- They have difficulty accounting for the multiplicity of the work environment. Increasingly flexible response to the differing environments is required. Traditional apprentice learning is encapsulated in the social practices of a discrete community whose relations with other social practices are much more regularised and stabilised. Flexibility, reflectivity and critique are necessary in ICT, not simply the uncritical socialisation in a single social practice.
- They have great difficulty accounting for the dialectical contradictions when apprentices have greater expertise in some areas that masters. The complex division of labour and cognition is not catered for by the apprenticeship model (Russell, 1998).
As the majority of student feedback was collected anonymously, it was not ethically acceptable to relate specific comments to particular student learning styles. An interesting insight may have been provided by the comparison: were particular learner types more/less comfortable with the Apprenticeship model (as the literature suggested), or were the comments across type? While this cohort’s profile indicated the learning model would not inhibit their learning, raw course results suggested most students were not interested in exploring beyond the set requirements of the course – student comments regarding not being pressured to complete weekly tasks support this perception. This implied that, while the model enabled effective learning across the learning styles, the motivational element was missing.

**Intervention 2: Adapting PBL**

The conclusion reached after the first intervention was that the master/apprentice relation could be down-played so that students took early control of their own learning rather than relying on the teacher/master to direct them in what they needed to learn, and that a more open approach to describing the design of the course might be beneficial to students challenged by its non-traditional nature. The second intervention, therefore, focussed on student-centred learning: creativity and adaptability within an explicitly PBL environment.

Student-centred is a term used to refer to learning environments that pay careful attention to the knowledge, skills, attitudes, and beliefs that learner brings to the educational setting (Bransford, Brown, & Cocking, 1999). This implies a need for students to assume a high level of responsibility in the learning situation and be actively choosing their goals and managing their learning, and involves considerable delegation of power by the teacher. This addressed the reliance on the ‘master’ found after Intervention 1.

While there are many views about the nature of creativity, there is some agreement that the creative process involves an application of past experiences or ideas in novel ways (Hennessey & Amabile, 2010). Common characteristics of creative individuals include ‘holistic thinking’ (in the sense they look for an overall broad thinking before moving into specific detail); proposing several candidate solutions early on (divergent thinking) in order to better examine the problem and help generate new concepts; a wide toolkit of domain relevant skills – and the ability to imagine/play out situations; creativity-relevant processes – including breaking perceptual and cognitive set and breaking out of performance ‘scripts’, suspending judgement; knowledge of heuristics (e.g. use of analogies, when all else fails try something counter-intuitive); adopting a creativity inducing work style (e.g. tolerance for ambiguity, high degree of autonomy, independence of judgement); and intrinsic task motivation (Amabile, 1996; Guindon, 1989, 1990). Csikszentmihalyi (1996)’s analysis emphasised the social nature of creativity. Focussing on creativity addressed the reliance on the ‘correct’ answer found in evaluating Intervention 1.

McCracken (1997) suggests that a principled academic education teaches how to place new knowledge in context and then use it in multiple contexts - providing the practitioner with adaptability. This addressed the “uncritical socialisation in a single social practice” Russell (1998) noted as an issue with apprenticeship models and also reinforced the importance of transfer.

The overarching aim in this intervention, therefore, was to focus on creativity and divergent thinking (as an indicator of adaptability), 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 (divergent thinking). Investigation of learning approaches described in the topography developed by
Horvath et al. (2004) suggested a problem-driven approach as a feasible course environment for achieving the aim identified for this intervention.

An extensive literature exists on the applicability of Problem-based Learning where an expanding knowledge base makes it impossible to include all the knowledge required for the beginning practitioner in the undergraduate curriculum. PBL emphasises ‘learning to learn’ by placing great responsibility for learning on the learner. It involves teaching both a method of approaching and an attitude towards problem-solving, and is characterised by its flexibility and diversity, since it can be implemented in a variety of ways in different subjects and disciplines (Savin-Baden, 2000). PBL is also designed to integrate the subject knowledge students require in order to solve a particular problem and therefore study issues at a deep rather than surface level. In this environment, the teacher becomes a facilitator for learners who are actively involved in the learning process.

Other characteristics of a PBL environment include (Boud, 1985; de Graaff & Kolmos, 2003; Jonassen, 2002):

- **Learning is collaborative with a focus on communication and interpersonal skills**: As PBL tends to take place in small groups, students have to work cooperatively to achieve their collective learning outcomes, with their level of independence measured by their ability to work with others. Consequently, communication skills, collaborative skills and reflective/self-evaluation skills can also be developed (thus addressing components A and D of the Professional Capability Framework)
- **Students are self-directed**: They are required to take responsibility for their own learning, with a leaning towards self- and peer-assessment (component D)
- **Its problem solving requires the mental representation of problematic situations**: The problem space must be constructed, either individually or socially through negotiation (component C)
- **Active, systematic manipulation of the problem space is required for PBL problem solving**: The ‘problematic’ nature of the situation enables learners to accommodate when their current experience cannot be assimilated into the existing mental models they possess (component B).

Its supporters claim PBL results in increased motivation for learning, better integration of knowledge across disciplines and greater commitment to continued professional learning (Boud, 1985). Without losing the benefits of a situated context, PBL is seen to offer the flexibility to cater for a variety of learning styles, integrating the learning of content and skills in a collaborative environment that reflect how learners might use them in real life (Oliver & McLoughlin, 1999). It also appeared to address aspects of expertise (that learning beyond the initial stages may best be achieved through situational case studies with rich contextual information (Dreyfus & Dreyfus, 1986)) and wider engineering education issues relating to generic skills and life-long learning.

The PBL environment applied as Intervention 2 was augmented by creativity-enhancing strategies drawn from Edmonds and Candy (2002), to enable innovative approaches to problem-solving in students, and to redress student perception of the discipline. An additional focus addressed metacognitive strategies and reflection as an aid to transfer of the skills and knowledge learnt.

The value of metacognition is confirmed in the recurring findings from Scott’s work on applying the Professional Capability Framework (previously discussed). A focus on flexibility and productive thinking is also necessary, so that students learn to use past experience on a general level, while still being able to deal with each new problem situation in its own terms. Gott et al. (1993) posit that this adaptive/generative capability suggests the performer not only
knows the procedural steps for problem solving but understands when to deploy them and why they work, in effect is ‘expert’ in the use of them.

The problem environment was set up to provide students with an opportunity to deal with a complex professional problem by ‘living’ it, with the Web providing a richly developed context for role-playing in a blended mode (where online interaction and resource provision are supported by face-to-face activities). As McLaughlan and Kirkpatrick (2004) note, Internet-mediated roleplay-simulations can be designed to maintain effectively the interaction necessary for individuals and groups to work in a way that is truly collaborative rather than simply supporting distributed individual effort. The success of this approach has been reported in Engineering (McLaughlan & Kirkpatrick, 1999), while an analysis of student performance in such an environment suggests it supports student learning about alternative perspectives on problems and encourages transfer of learning to new contexts (McLaughlan & Kirkpatrick, 2004).

The development of the problem and its context was a considerable challenge: it had to be suitably complex and open-ended in order to instantiate the constructivist principles; it had to exhibit ill-structure so that students would engage with complexity, and present claims and rationale to negotiate understanding with their peers; it had to be too ‘large’ for individual students to complete successfully on their own; it had to address the learning outcomes identified, and provide a mechanism for students to demonstrate their competency in achieving these.

The scenario developed for the CreativePBL environment focuses on the secondment of the class to a (virtual) organisation – collaboration between a software house and the university. MurSoft required a team to work, on short-term placement, on a project to develop gaming software to be used as an educational resource within a tertiary institute. This provided an authentic context for learning: students were to have an opportunity, within their final year of study, to undertake an internship with a software-based organisation.

As well as providing a dedicated ‘office’ space – a laboratory specifically designated for use by SE students in third and fourth year exclusively, a collaborative on line work space was enabled for each group. Only group members and the teacher had access to this area – with students required to maintain all documentation on the site. This resource facilitated construction of the shared understanding necessary for collaborative learning, and enabled students to tune the accuracy and suitability of individual understanding through disentangling cognitive conflicts. It also introduced technical issues of version control and change management, also authentic in the domain. In addition, the representation tools provided (mindmapping, models through the CASE tool) acted as mediators for collaborative learning.

Students are expected to undertake the learning tasks collaboratively by means of the learning environment. All interaction with the ‘client’ was undertaken through Web-based material: a fictitious character, the Team Manager acted as go-between for the team and the client, while memos, minutes of meetings, telephone messages, ‘talking heads’, press releases etc. provided the triggers required by the PBL process. These acted as prompts to students to undertake some task identified in the problem. Support material was provided, as were expert consultants (either from industry or the School) as required. The lecturer was always available as a facilitator.

Course content was centred on the online teaching material (now viewed as a resource repository in the problem environment) within the Software Factory previously described, and a recommended text. These acted as a constraint: students initially explored this material in order to achieve the learning outcomes they have identified in a problem component, rather than having unlimited access to resources on the Internet and elsewhere. On the other hand, it was important that students became aware that other views existed. Again, providing
environment constraints added to the authentic approach: as graduates, students would be expected to follow the operating procedures standardised within the employing organisation, rather than having the freedom to choose those which suited them individually. The Web material was expanded to include resources on the PBL methodology, including support documents for students on the PBL process, self-assessment items and trigger checklists (as triggers (the memos etc. previously mentioned) were released automatically).

The decision was taken to minimise technical complexity (e.g. in terms of advanced interactive media and ‘appearance’) so that students could easily access the resource off-site and with minimal software requirements (a Java runtime environment and Flash reader). Pragmatically, technological capacity could never keep up with student expectations (in general males in their late teens, very au fait with gaming environments). Rather, ‘cognitive realism’ (Herrington, Reeves, & Oliver, 2003) to the real-life task (as opposed to a technological-driven view) was seen as having (much) greater significance.

Students were expected to construct a problem representation based on the triggers provided, and manipulate the problem space so that an external representation (in this case the artefacts that were deliverables) could be created from the multiple individual representations at a fine enough grain to demonstrate group and individual achievement of the appropriate learning outcomes. The course still focussed on RE in an object-oriented context.

As with the Cognitive Apprenticeship environment, evaluating the success of the CreativePBL intervention was based on strategies in several dimensions: the success of the implementation of the PBL model and the effect of the intervention on the development of the student cohort, the latter in terms of both performance in assessable tasks and their perceptions of learning in this environment. Throughout the semester each problem required use of the products of the preceding problem. In this way, students were able to activate (recent) prior knowledge in order to commence the PBL process for the next problem. Since rework was required to redress issues identified by their ‘manager’, any re-conceptualisation undertaken by the students (after feedback or when tackling a subsequent problem) was reflected in their increased understanding of the context (addressing components B and C of the Professional Capability Framework).

In order to facilitate learning in this environment three ‘processes’ were made explicit to the students:

- **The PBL process:** That advocated by Koschmann, Myers, Barrows, and Feltovich (1994) was used to anchor the student within the learning environment, and provides the discipline to assist in content learning. The first two workshop sessions were structured to introduce PBL as an explicit process with defined phases, provide an introduction to the role-playing environment and complete a small problem with the teacher, by ‘visiting’ each PBL phase. In effect what was offered was a mini-apprenticeship, with the process modelled and discussed. Support documentation for PBL was also made available through the online environment.

- **Approaches to Learning:** Students were exposed to a variety of personality and learning style instruments (e.g. Keirsey and Bates (1984), Kolb (1995), Soloman and Felder (1999), Entwistle and Ramsden (1983)). The purpose was to provide strategies they could apply to help overcome any dissonance between their preferred method of learning and the learning environment, and to assist in dealing with the team dynamics encountered.

- **Expertise:** Students were introduced to the ‘process’ for becoming expert in a dis-
cipline. This involved exposure to learning strategies that model expert learning (see Donald (2002) for a discussion on how people think and students learn in specific disciplines), including the ability to: manage knowledge structures (to plan) and exploit opportunistic and creative cognitive behaviour; collect, manipulate and analyse many different forms of data and then present them in meaningful ways; manipulate multiple representations of problem components and to effectively utilise the notations and symbol systems that are shared knowledge. In effect, exposure to cognitive flexibility (Spiro, Feltovich, Jacobson, & Coulson, 1991).

The results of this intervention indicated the PBL environment was also achieving some of the goals hoped for: students perceived themselves as more confident of their skills, and more willing to be innovative. However, they also raised issues about the relevance of individual components of the course (in particular non-problem-related items such as those included in the portfolio, which required reflection on individual and group achievements during the session), suggesting generic skills and reflection were still not considered important. Nevertheless, over 65% of the students expended effort to pass this component (up from 50% the previous year), with 41% of students (up from 12%) submitting well beyond the requirements of the course content. Despite some negative comments about the learning model (don’t really like how it’s structured; does not have proper course structure; no lecture or tutorial) other students acknowledged its value (makes you think; helps with thinking about all areas of a problem (good for other units)), and considered the course well structured; interesting; practical; well presented; good for software industry; it’s really good.

Students perceived they learnt more in the areas of research, communications (confidence to speak up; need to be heard & get ideas across) and team skills. They added: concepts easier to grasp; forced to learn more for project relevant components and, finally, they had to grapple with various perspectives from others. These student comments would indicate many components of the Professional Capability Framework had been successfully addressed.

Of major concern, when evaluating this intervention, was the realisation that the students were very product oriented – they saw the artefact (generally the code they developed) as the primary goal of the activities they undertook. Being made to focus on process to (in their perception) the detriment of the product was very frustrating, and had some negative effect. The PLB focus on process was ultimately in conflict with the aim of the intervention – to model professionals in practice. Schön (1983) notes that in the ordinary form of practical knowledge practitioners do not think about what they are doing, except when puzzled or surprised. He named this reflecting-in-action, and argued that it is central to the ability to act effectively in unique, ambiguous, or divergent situations.

**Intervention 3: Studio Learning**

The third intervention was developed to gain leverage from the positive elements of the both the apprenticeship model and the CreativePBL environment, while addressing the negatives. The PBL environment did not allow for a ‘master’ role, and therefore ‘lost’ the expertise the teacher provided: there the teacher acted as facilitator instead. The apprenticeship model that featured ‘fading’ as students gained competency, in contrast, allowed the students to depend on the ‘master’ extensively. Therefore, in the same way that the master/apprentice model addressed some aspects of discipline practitioner action and inhibited others, the facilitation aspect of the PBL model also exhibited elements of an ‘incorrect’ learning environment. A mentor/protégé relationship allows teacher and learner to seek to understand each
other’s position with the aim of agreement and/or defensible deviations. However, this requires a confidence on the part of the learner that is not often present at novice stage, and therefore needs to be fostered. The work of Laurillard (1993) develops this concept of learning as a dialogue.

Some learning models for adaptive and flexible learning (and, by implication, supporting soft skill development) are based on reflection: Kolb (1984)’s work presents learning as a process in which knowledge is created by the learner through some transformation of experience; Schön (1985) refers to reflection-in-action as the responses that skilful practitioners bring to their practice. This reflection consists of strategies of action (ways of framing the situations encountered in day-to-day experience), and may take the form of problem solving, theory building, or re-appreciation of the situation. The success of this learning is based on factors such as the degree of learner control, degree of correspondence of learning environment to real environment and degree of involvement of self and the adoption of strategies which have come to be identified as contributing to reflection. If students are being prepared to become reflective practitioners (Schön, 1983), opportunities for students to develop reflective skills and sensibilities should be embedded as a normal part of all professional courses (Boud, Keough, & Walker, 1985; Schön, 1987).

As noted previously, the approach to learning that a student takes is sensitive to the context in which learning is done, with a deep approach to learning required to develop advanced conceptions. Learning diagnostics (e.g. ASI) undertaken as part of Intervention 2 showed there was as strong a bias to surface learning as there was to deep learning in the student cohort. This related (and had impact on) the critical concept of lifelong learning. Therefore metacognitive strategies needed to be made explicit – students required a rationale for including reflection as an element of their learning (demonstrated through portfolios and activity logs). Edwards (2004) reports on approaches to explicitly provide the opportunity for students to adopt expert strategies. These enable the teacher to guide students in the nature of expert processes and help them to reflect critically on their effectiveness. He notes that the best means of facilitating expertise is to provide the opportunity for practice. However, only through encouraging students to challenge their own effectiveness can they learn what this implies.

What was needed, then, was a model of education that added, to the positive aspects of the studios as an immersive environment, a focus on metacognition, so that learning integrated evaluation of the ‘practical’ outcomes of the problem with the creative process. To Laurillard (1993)’s learning model (where attention on key relationships between forms of discourse, academic knowledge, interactions with the world and reflection are emphasised) aspects of teachback and self-explanation were incorporated, forcing a focus on key aspects of the domain, deeper processing of the topic, and allowing failures and conflicts to emerge (Gobet & Wood, 1999).

These ideas formed the basis for the StudioLearning model developed and applied in this final intervention. The StudioLearning model allowed for approaches from previous interventions to be integrated. The teacher acted as expert consultant, and could be ‘engaged’ by the students to provide modelling demonstration and domain expertise. However, the students were still required to direct their own learning: developing the learning objectives to address each problem presented within the PBL environment previously developed. Here the strategy was to reach all types of learners by ‘teaching around the cycle’ (Kolb, 1984), thus enabling students to develop the mental dexterity required in professional practice, and emphasising the importance of contingency measures, opportunism, reflective practice and metalearning. This StudioLearning environment also provided the opportunity for students to adopt expert strategies.
Engaging Software Engineering Students with Employability Skills

– the teacher acted as guide/mentor or ‘consultant’ in these processes, helping students to reflect critically on their effectiveness in specific contexts.

Kuhn (2001) describes characteristics of the studio environment:

- Student work is organised primarily into complex and open ended problems
- Students’ design solutions undergo multiple and rapid iterations
- Students are exposed to relevant precedent and to rapid iteration of design solutions. This combination is seen as essential to the development of true expertise (Dreyfus & Dreyfus, 1986)
- Critique is frequent, and occurs in both formal and informal ways, from teachers, peers, and visiting experts. One of the hallmarks of studio education is the creation of a ‘culture of critique’, in which students, who spend long hours working side by side at their projects, give each other frequent feedback, and also get both formal and informal feedback from the academic in charge of the studio in order to reflect on their learning
- Students study precedents (past designs) and are encouraged to think about the big picture
- Teachers mentor students to impose appropriate constraints on their design process in order to navigate a complex and open ended problem and find a satisfactory design solution
- The appropriate use of a variety of design media over the course of the project significantly supports and improves students insight and designs.

A studio environment has been applied to SE education, most famously at Carnegie Mellon (Tomayko, 1996) as part of their Masters’ programme. In undergraduate education, studios are more often applied to capstone courses rather than foundational learning (Ramakrishnan, 2003), although recent success in this area has been reported (Narayanan, Hundhausen, Hendrix, & Crosby, 2012).

At this time, a decision was made that all 3rd and 4th year learning within the School (in effect the final two years of undergraduate engineering degrees) would apply a studio model, with two studios considered a full-time load. Later evaluation indicated the appropriateness of this approach. Students noted that with all their studies undertaken within studios, they felt they were much more in control of their efforts:

- Academic staff were perceived as more tolerant of the needs of other studios
- With a full-time load of only two studios student time was not as fragmented across different areas
- Except for the (negotiated) compulsory attendance, students could vary the time they spent on each studio in response to their total learning context. It was the team’s role to ensure tasks were on schedule. They concluded that this flexibility reduced stress and allowed them to focus on the learning they needed to achieve for the task.

In order to encourage deeper learning, a portfolio (worth 20% of the final mark) comprising, as its major element, the concept maps for the topics explored throughout the course, was included as one of the artefacts developed throughout the semester. The maps were evaluated on a regular basis by team members. This peer review was seen as a mechanism to assist the development of a common understanding of the knowledge being developed within the course – simply, as a view into the minds of the students (Freeman and Urbaczewski, 2001). The portfolio was now both a group and individual effort and included a reading log compiled by the group (but initialled
Engaging Software Engineering Students with Employability Skills

by whoever had written the particular summary), minutes of group meetings throughout the semester (with rotating minute taker) to indicate task allocation, and individual activity logs for ‘in- and out-of-class’ time spent on the course. For this component, students were required to indicate time spent on tasks such as: discussion (time spent discussing issues internally within group); consultation (time spent discussing issues with teacher); research (time spent researching new areas of knowledge to assist with engineering of project); design work (time spent on formulating and describing design solution); review & testing (time spent on reviewing the design solution and/or testing it); personal management (time spent on individual management, documentation and related tasks); group management (time spent on project management related tasks for the group as a whole). Students also answered several questions on a weekly basis: what did I achieve this week? what issues did I have this week? what can I do next week to address these issues?

The evaluations of this intervention focussed on the level of success of StudioLearning in relation to its implementation, and on the effect of the intervention on the development of the student cohort - examined in terms of short term impact (performance of the students in assessable tasks during the intervention) and longer term impact (performance of the students in assessable tasks dependant on the learning objectives of the intervention, in the subsequent course).

For the discipline of SE the StudioLearning model appeared to hold the most promise – students had been provided with a process during an orientation week at the commencement of semester, and were able to apply it in the course, but also adapt or discard it as they perceived necessary. In this way, the environment supported both discipline and opportunism. The student data indicated they were taking control of their own learning and acknowledging the importance of skills such as time management: (while I kept up with the readings and assignments I ended up getting behind in creating the mind maps and logging my personal journal. While I acknowledge that is my responsibility to manage my own time it would have been useful if this task was checked more frequently). Robillard (2005) noted that, in a typical opportunistic problem-solving activity, individuals spontaneously rely on teammates to provide missing information. Evaluation of the StudioLearning model in relation to the Professional Capability Framework shows this was occurring, and that other components of the Framework were being addressed:

Some of the knowledge I have learnt has resulted from the interaction with my team members i.e. I don’t believe the level of understanding I now have, would have been achieved by working on the assignments by myself (component A)

we have to apply learning to a realistic problem which means it moves as out in the real world e.g. the lecturer pointed out errors in thinking and this resulted in us having to revise what we had completed previously in order to move to the next step. I found this gave me a greater depth of knowledge than the usual do an assignment get some of it wrong and move on to the next usually non related assignment; this method of teaching has provided me with a frame work that I can use to identify future problems and develop solutions (component B)

each of us shares the ability or outcomes from the ability e.g. ability to research (find information/knowledge), understanding a problem that others don’t and interpreting it into a context the other can understand so they can solve the problem (component C)

the personal journals can be a useful tool to ensure the unit learning’s are integrated into the students existing knowledge/experience and that the students behaviours are modified to align with those required to be a successful engineering
graduate. [...] weekly journals would be useful in ensuring they are being used as a tool to guide student development into the higher levels of Bloom’s taxonomy and the non-technical aspects of their learning; this unit teaches a process that is built on knowledge but more importantly that knowledge is converted to a skill via practice on the problem. I don’t believe this is achieved by the other style of teaching e.g. lectures and exercise type assignments (components D/E).

The portfolio entries showed that students were highly motivated to complete the tasks assigned – the suggestion is that general student attitude towards the controllability of the learning outcome (i.e., externally dictated and beyond student control, or within the internal control of the student through effort and personal interest) influences their motivation and the level of achievement in the learning process. In addition, motivation is heightened in situations which offer opportunities for increasing competence or intelligence. More interestingly, as students rotated into the role of Project Manager, they (individually) applied what they had previously learnt with regards to learning strategies and approaches to study in order to motivate their group members.

Observation and analysis of learning of some of the cohort in the subsequent course showed strong indications of willingness to transfer knowledge gained, to take control of their learning, and indicated motivation to deeper learning. Examination of student reflective comments, in conjunction with data regarding student learning, added another dimension to the issue of education for competent practice. This examination indicated a relationship existing between the learner and the learning model, so that students whose approach favoured deep learning for understanding were advantaged by a learning environment which challenged them (I have noticed that the design studios require a lot more work from me than if I was working alone. For example I have to spend more time working on problems because of the extra overhead of working in a team (meetings and social interaction). There is also the need to do extra research to gain information that is normally just handed out in a lecture. However I don’t mind putting in the extra effort because I feel the extra effort is worth it because I feel more confident that I do know the material (not an impostor) and can apply it to future situations). Use of the term ‘impostor’ is perhaps telling.

**STUDENT DEVELOPMENT ACROSS INTERVENTIONS**

Grow (1996) describes an approach to modelling learning from the learner’s ‘growth’ towards lifelong learning. This model reflects the principles advocated in student-centred learning environments: the learner determines the need for some education, decides on a preferred approach to learning, identifies and accesses learning resources and draws on the assistance of educators as a part of that overall strategy rather than as a central element:

- **Stage 1**: Learners need an authority figure to give them explicit directions on what to do, how to do it, and when. They either treat teachers as experts who know what the student needs to do, or they passively slide through the educational system, responding mainly to teachers who ‘make’ them learn. The teacher acts as authority, coach.
- **Stage 2**: Learners are ‘available’. They are interested or able to be interested. They respond to motivational techniques. They are willing to do those assignments they can see the purpose of. They are confident but may be largely ignorant of the subject of instruction. The teacher acts as motivator, guide.
• **Stage 3:** Learners have skill and knowledge, and they see themselves as participants in their own education. They are ready to explore a subject with a good guide. They will even explore some of it on their own. But they may need to develop a deeper self-concept, more confidence, more sense of direction, and a greater ability to work with (and learn from) others. These learners benefit from learning more about how they learn, such as making conscious use of learning strategies. The teacher acts as **facilitator**

• **Stage 4:** Learners set their own goals and standards – with or without help from experts. They use experts, institutions, and other resources to pursue these goals. Learners at this stage are both able and willing to take responsibility for their learning, direction, and productivity. They exercise skills in time management, project management, goal-setting, self-evaluation, peer critique, information gathering, and use of educational resources. The teacher acts as **consultant, delegator.**

In summary, Intervention 1 showed that, even within a constructivist framework, the relationship between teacher and learner (or master and apprentice) can remain unidirectional—the former modelling behaviour for the latter and guiding attainment of the learning outcomes (Grow’s Stage 1). Intervention 2 provided an environment for students to address Stages 2-3—the PBL environment required a more explorative and collaborative approach to learning. Finally, Intervention 3 supported a transition to Stages 3-4. Students were able to demonstrate reflective-practitioner behaviour, and were comfortable (and successful) in an environment where the teacher was not the authority but one of the resources they had access to.

**SUPPORT FOR SOFT- SKILLS LEARNING**

The results of this project show that a critical aspect of the challenge is integrating non-technical skills across the whole curriculum, so that students do not receive conflicting messages about their importance. Making acquisition of these skills explicit (by, for example, addressing specific soft skills learning outcomes, activities and assessment of them) was valuable. Finally, requiring students to incorporate metacognitive strategies and reflection in the evaluation of their own learning enabled students to track and acknowledge their increasing competence in these skills.

Implementing environments such as those described in this chapter requires a cultural change to occur within the educational environment for the discipline. Not only do students need to adapt their learning behaviour so that a reliance on lecture/tutorial/laboratory is minimised (if not removed totally), but academics also require orientation to less traditional learning models. In this context, although the interventions were ‘public’ within the School, and reported on within School seminars, in general it could be said that academic staff were not deeply conversant with the approaches described. Therefore a series of staff development workshops was initiated for academics in order to provide training and education on problem- and studio-based learning.

The staff development workshops were structured to address two separate concerns:

• **Provide a background in PBL and studio approaches by undertaking a PBL session:** The outcome of this was an understanding of what changes would be needed to enable studios across all courses, and a set of tasks to be completed by each academic staff member for each of the courses they coordinated. Tasks included
Engaging Software Engineering Students with Employability Skills

curriculum mapping and problem development, amongst others. Concerns were also raised – the most critical being discipline content coverage. In addition, several staff members were not convinced that basic (ie foundational) knowledge could be learnt this way.

- Develop an appreciation of the issues raised by such active learning: The School invited an academic who had a great deal of experience in applying PBL in an engineering context. The result of this workshop was clearer commitment on the part of academic staff to make studios work.

Students were also provided with exposure to the learning environment prior to semester start. A one week orientation programme, in which students were placed into multi-(engineering) discipline teams, provided the opportunity to model StudioLearning and establish the roles and responsibilities of students and academics within this model. These objectives were achieved through a small-scale design task as a means of identifying and exposing the StudioLearning approach. The Design Week provided an introduction to generic tools, techniques, methods and processes as support services made available with the learning environment. These might otherwise have needed to be duplicated in each studio. This orientation also provided an opportunity to pre-test student perceptions of the model.

The adoption of StudioLearning across the School may be considered ‘transformational’, a second order change that required a fundamental shift in behaviour and resources (Levy, 1986). A correlation appeared to exist between attendance at staff development sessions and the ‘success’ of the specific studio. The implication of this result is that no staff should teach within a studio environment without adequate and appropriate training (both in the learning theory behind the model and in facilitation techniques). Therefore, while it is clear from the literature that an understanding of the underpinning pedagogical basics is necessary, as are special facilitation skills, it has resource implications within the tightening finance environment prevailing at many universities.

CONCLUSION

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. The interventions described in this chapter addressed 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’ in their team, 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)
Engaging Software Engineering Students with Employability Skills

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 open to discussion and feedback and willing to retrace their steps/redo the work in order to advance to a solution; they 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, within an environment that enabled the advantages of ‘flow time’ to be exploited

- **Provide challenges:** Students were motivated by the (increasing) complexity of the task, and were able to focus on cognitive and interpersonal skills to adapt to the changes required.

While technical know-how acts as the gatekeeper to professional practice, the non-technical, ‘soft’ or ‘employability’ skills are those which provide career resilience (Waterman et al., 1994). Ensuring graduates are ready to apply these as practitioners remains a challenge in higher education.

**REFERENCES**


Engaging Software Engineering Students with Employability Skills


Engaging Software Engineering Students with Employability Skills


Lethbridge, T. C. (2000). What knowledge is important to a software professional? IEEE Computer, 33(5), 44–50. doi:10.1109/2.841783


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


Engaging Software Engineering Students with Employability Skills


Engaging Software Engineering Students with Employability Skills


**ADDITIONAL READING**


Engaging Software Engineering Students with Employability Skills


Engaging Software Engineering Students with Employability Skills


**KEY TERMS AND DEFINITIONS**

**Active Learning:** An umbrella term that refers to students learning by engaging in activities (as opposed to passive listening). Activities can include reading, writing, problem-solving, reflection, often in a collaborative setting.

**Cognitive Flexibility:** The ability to represent knowledge from different perspectives, switching to accommodate changes in the environment that knowledge applies to. This is related to the transfer of knowledge and skills beyond the initial learning environment.

**Community of Practice:** A group with a common interest seeks to learn from each other through and participating in shared information and experience.

**Constructive Learning:** The learner constructs knowledge based on an inconsistency between current knowledge and experience, usually undertaken in a social context.

**Employability Skills:** Within an Australian context, employability skills have been identified as those non-technical skills that enable successful workplace performance: communication, team work; problem solving; initiative and enterprise; planning and organising; self-management; learning; technology.

**Higher Order Thinking:** Learning taxonomies suggest some learning is foundational (to know, comprehend, apply) while other learning is a complex combination of these skills, and requires critical and abstracting abilities. These (to analyse, synthesise, evaluate) are considered higher order.

**Metacognitive Skills:** Involves a learner’s awareness of their own learning process (rather than the content of the learning). The learner monitors self-awareness and uses it, implying higher order thinking, to control and improve the learning process.