



Higher Education Research and Development Society of Australasia, Inc

Learning for an Unknown Future

Proceedings of the

26th HERDSA Annual Conference

6-9 July 2003

Christchurch, New Zealand

Armarego, J. & Clarke, S. (2003) Preparing students for the future: learning creative software development —setting the stage, in *Learning for an Unknown Future, Proceedings of the 26th HERDSA Annual Conference, Christchurch, New Zealand, 6-9 July 2003: pp 17.*

Published 2003 by the
Higher Education Research and Development Society of Australasia, Inc
PO Box 27, Milperra, NSW 2214, Australia
www.herdsa.org.au

ISSN: 0155-6223
ISBN: 0 90 8557 55 8

This research paper was reviewed using a double blind peer review process that meets DEEWR requirements. Two reviewers were appointed on the basis of their independence, expertise and experience and received the full paper devoid of the authors' names and institutions in order to ensure objectivity and anonymity. Where substantial differences existed between the two reviewers, a third reviewer was appointed. Papers were evaluated on the basis of originality, quality of academic merit, relevance to the conference theme and the standard of writing/presentation. Following review, this full paper was presented at the international conference.

Copyright© 2003 HERDSA and the authors. Apart from any fair dealing for the purposes of research or private study, criticism or review, as permitted under the Copyright, Design and Patent Act, 2005, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers at the address above.

Preparing students for the future: learning creative software development —setting the stage

Jocelyn Armarego

Murdoch University, Rockingham, Australia

jocelyn@eng.murdoch.edu.au

Sally Clarke

Murdoch University, Rockingham, Australia

***Abstract:** One of the challenges being addressed within education, and higher education in particular, is that of providing students with life-long learning skills. The speed with which technology evolves, the multiplicity of its impact on society and the ramifications of that impact mean that more than technical competence with specific tools and techniques is necessary. This is especially true of disciplines where changing technology is one of the *raison d'être*. This paper focuses on the background to an approach based at the School of Engineering, Murdoch University to enhance the adaptability skills and knowledge of Software Engineering students by stressing the need for divergent thinking and creativity within a formal course in Requirements Engineering.*

***Keywords:** software engineering education; creativity; problem-based learning*

Introduction

The nature of software development

Software development has been described as a craft. The negative connotations of this label include an inability to consistently guarantee a quality product, fit for the purpose for which it was developed, produced on time and within budget. A study of over 8,000 projects (Standish, 1995) reported 16.2 percent of software was successful, 52.7 percent were over budget, time and had fewer features, and 31.1 percent of projects were cancelled. These rates do not significantly differ from those reported in the 1970s and 1980s (Mann, 1996). Many of the shortfalls may be traced to deficiencies in formulating a description of the system to be developed, which is the subject matter of Requirements Engineering (RE), and the focus of this paper.

Royce (1970) was the first to note explicitly that an engineering approach to software development was required, in the expectation that adhering to a defined, repeatable process would enhance quality. The underlying assumption is that the world works rationally and that therefore 'good' software development is achieved by applying scientific investigative techniques (Pfleeger, 1999). This focus on engineering is mirrored in the education of software developers. Where IEAust (Institution of Engineers, Australia) accredited two undergraduate programs for the engineering of software in the mid-1990s (Melbourne,

Murdoch), by 2002 this figure approached 20. A similar trend is being shown in the US, with an exponential growth in offerings of undergraduate software engineering degrees. However, recent work (Maiden & Gizikis, 2001; Nguyen & Swatman, 2000) argues such an approach should be regarded as flawed. Software is a collaborative invention: its development an exploratory and self-correcting dialogue (Bach, 1999).

There are positive implications as well for the label 'craft'. Each system is considered a unique synergy between the hardware, software and organisational context in which it will be utilised. This approach suggests that the development process cannot be repeatable, as the forces at play will differ for each context, continually changing as understanding of the characteristics of the developing system grows in all stakeholders. In this alternate perspective the process of RE seen as one of insight-driven knowledge discovery (Carroll & Swatman, 1999; Guindon, 1989) facilitated by opportunistic behaviour (Guindon, 1990; Visser, 1992). Participants in the process must remain sensitive to progressive modifications (Gigch, 2000) which lead not to a problem-solution, but to an 'evolved fit' acceptable to all stakeholders within the problem space.

The quintessential *creativity* of this process (Lubars, Potts, & Richer, 1993; Maiden & Gizikis, 2001; Maiden & Sutcliffe, 1992; Thomas, Lee, & Danis, 2002) is hampered by strict adherence to engineering and science methodologies. These:

- restrict essential characteristics of the process such as opportunism (Guindon, 1989);
- assist in adding accidental complexity through their attempts to control the RE's professional practice. (Sutcliffe & Maiden (1992) suggest strict adherence to methods and procedures may restrict natural problem-solving); and,
- impose a plan at odds to the RE's cognitive planning mechanisms and hence interfere with the management of knowledge. (Visser & Hoc (1990) suggest that, in practice, a plan is followed only as long as it is cognitively cost-effective).

Educating software developers

Approaches to training Requirements Engineers based on traditional learning models tend to emphasise technical knowledge, and are based largely on notations and prescribed processes. This is at odds with the inherent characteristics associated with real requirements problems, where:

- complexity is added to rather than reduced with increased understanding of the initial problem;
- metacognitive strategies are fundamental to the process;
- problem-solving needs a rich background of knowledge and intuition to operate effectively;
- a breadth of experience is necessary so that similarities and differences with past strategies are used to deal with new situations. (Bubenko, 1995)

Glass (1995) suggests that discipline and creativity are the *odd couple* of software development—the discipline imposed by methodology, for example, forms a frame for the opportunistic creativity of design. The educational dilemma becomes one of providing an educational base that enables software developers to both create and engineer the systems they build: to be adaptable to the changing environment that is inevitable in their chosen discipline.

A Problem-based learning (PBL) approach promises to provide students with a solid foundation in subject matter while at the same time exposing them to real-world characteristics. It provides students with a process to deal with problems within a metacognitive-rich framework that makes complexity apparent and lets students deal with it explicitly.

The place of creativity

Thomas et al. (2002) suggest there is a widening gap between the degree of flexibility and creativity needed to adapt to a changing world and the capacity to do so. They suggest that difficulties can be attributed to the following:

- individuals or groups do not engage in effective and efficient processes of innovative design. As examples of *structuring failure*, people typically fail to spend sufficient time in the early stages of design: problem finding and problem formulation, then often bring critical judgment into play too early in the idea generation phase of problem solving. As another example, empirical evidence shows that people's behaviour is path-dependent and they are often unwilling to take what appears to be a step that undoes a previous action even if that step is actually necessary for a solution (Thomas, Lyon, & Miller, 1977);
- individuals have a large amount of relevant *implicit knowledge* they often will not bring to bear on a problem. Providing appropriate strategies, knowledge sources or representations can significantly improve an individual's effectiveness in problem solving and innovation (Thomas et al., 1977)
- the appropriate level, type, and directionality of motivation are not brought to bear (Amabile, 1983).

Amabile's general theory of creativity suggests that three components are necessary for the enhancement of creative potential. They are:

- domain relevant skills—the more skills the better, and the ability to imagine/play out situations;
- creativity-relevant processes—including breaking perceptual (the way you perceive a situation) and cognitive (the way you analyse) set and breaking out of performance 'scripts', suspending judgement, knowledge of heuristics, adopting a creativity inducing work style (eg tolerance for ambiguity, high degree of autonomy, independence of judgement);
- intrinsic task motivation.

These components are critical in the PBL-based learning environment we are attempting to provide within the School of Engineering Science at Murdoch University.

The relationship between creativity and instruction has been a focus of research. Formal education emphasises logical rather than divergent thinking, well-defined problems, good grades, and values conventional behaviour (Albert, 1996). In addition, many cultures (here we may say discipline-based as well as social) encourage respect for the past and discourage disruptive innovations. Promoting widespread creativity raises expectations that may change employment patterns, educational systems and community norms.

Patel, Kinshuk, & Russell (2000) argue that learners in a traditional setting predominantly constitute students preparing for a career—classroom based students. With the relevance of domain knowledge not fully understood by students, the focus shifts to skills that will yield

higher grades as an immediate objective. 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, thus defeating the objectives of providing a well-balanced learning experience. These characteristics are at odds with the learning objectives of the RE course as well as with current higher education graduate goals.

The Murdoch context

The School of Engineering Science provides a four-year undergraduate Software Engineering (SE) degree. Requirements Engineering is the first of the core SE courses, offered in semester 1 of the second year of study. Students have generally been immersed in a scientific/engineering paradigm. Laboratory procedure, repeatability of experimentation and rigour in mathematics are the key learning objectives of the introductory courses taken.

RE provides a contrast that some students find difficult to assimilate. Although due process and procedure have their place, the focus of the course is on divergent thinking and the development and evaluation of alternatives. Students come to the course with some competence in programming. In Requirements Engineering they are asked to ignore the (coding) solution to a situation presented, and to explore and then formulate the problem itself. Experience in teaching RE has shown that students' expectations are challenged, in that:

- they expect there to exist a definitive solution to the problems with which they are presented (à la science/mathematics);
- they expect to define the problems only in terms of the programming language with which they are familiar (currently Java);
- they expect a fundamentally competitive class environment to exist;
- they expect their 'wild ideas' to be laughed at and ultimately rejected, and therefore are inhibited in expressing them.

Since its inception in 1999, the course has been taught in workshop mode. All material is available online, so lectures and tutorials are replaced by discussion, exercises and group evaluation of alternatives presented. Academic results and students' evaluation of teaching suggest that the course is successful. However, a review of the course based on Reeves (1997) showed that RE had a reasonably high level of teacher direction. These observations were borne out by students' responses to changes made to a follow-on course (Advanced Software Design II (ASDII)) presented in a learner-centred mode. The expectations noted above were still evident (Figure 1 shows the evaluation results for both courses).

Therefore, the prime motivation in changing the existing learning environment was to address these issues as early as feasible. It also seemed that the false expectations may be more easily challenged through less traditional approaches to learning, where the focus is on learning in a collaborative environment, with emphasis on 'learning to learn' and the placing of greater responsibility for learning on the learner (Wilson & Cole, 1996).

The attributes of a problem-based learning classroom (Boud, 1985) provide a framework for future learning (Wilson & Cole, 1996). In addition, supporters of PBL argue that it provides an effective environment for future professionals who need to access knowledge across a range of disciplines. The positive influences of an appropriate environment on the development of creative potential also support the adoption of PBL for RE education. According to Amabile (1996) these include: encouraging the assertion of ideas; no reliance on

order and training; no fear of failure; providing time and resources; developing expertise, giving positive, constructive feedback that is work or task focussed; encouraging a spirit of play and experimentation providing a mix of styles and backgrounds with opportunities for group interaction; making a safe place for risk taking; allowing free choice in task engagement; offering rewards that recognise achievements or enable additional performance but maintain intrinsic motivation rather than controlling behaviour.

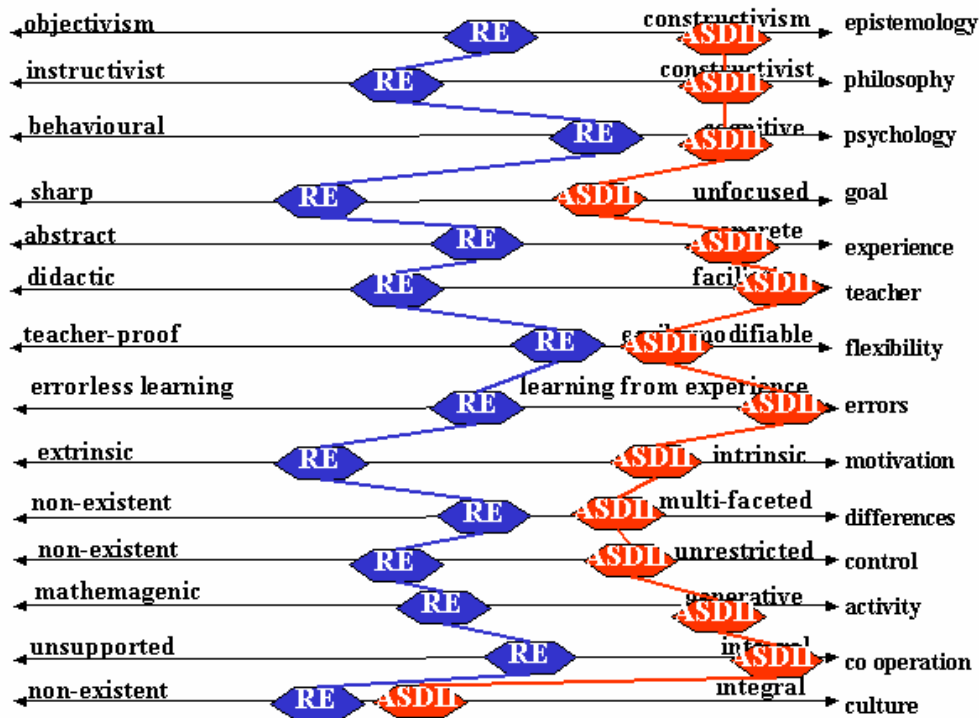


Figure 1: Evaluation of RE and ASDII using Reeves' (1997) framework

A framework for creative RE education

The RE-PBL world

The Murdoch RE course material has been reworked for a PBL environment, and taught in this mode from February 2003. A PBL process is used to anchor the student. The congruence between Edmonds and Candy's (2002) elements of creativity and the PBL stages enables creative activities to be embedded into the PBL process (see Table 1). This creative PBL process addresses the issues highlighted by Thomas et al. (2002):

- *problem analysis* is a critical stage: starting from the unknown and progressing to a description of the problem itself, and the knowledge needed to deal with it. Problem-solving habit is challenged by the need to generate alternate solution paths. In the RE course this problem analysis is a critical outcome;
- the value of alternative perspectives is fostered through participation in a collaborative environment and the active promotion of *critical friendship*. Critical appraisal and self appraisal skills are developed through the use of reflection tools such as the 4SAT (Zimitat & Alexander, 1999);
- although external motivation is difficult to eliminate within an undergraduate degree, PBL is seen to foster *intrinsic motivation* through the authenticity of the tasks

undertaken (Wilson & Cole, 1996). Emphasis is placed on constructing a framework in which details of the problem are situated.

Table 1: PBL process and creativity activities

PBL Stages (Koschmann, Myers, Barrows, & Feltovich, 1994)	
<i>problem analysis:</i>	the rich context is mined for important facts, sub-problem(s) and alternate solution paths generated
<i>self-directed learning:</i>	the learning agenda is determined by the information needed to evaluate the alternatives proposed
<i>problem re-examination:</i>	based on findings, solution paths are added, deleted or revised
<i>abstraction:</i>	an articulation process to increase the utility of the knowledge gained in specific contexts
<i>reflection:</i>	a debriefing of the experience to identify improvement in the learning process.
Creative activities (Edmonds & Candy, 2002)	
<p><i>exploration</i> of ideas, knowledge, and options, based on:</p> <ul style="list-style-type: none"> ▪ <i>breaking with conventional expectations</i>, whether visual, structural, or conceptual, is a key characteristic of creative thought ▪ <i>immersion</i>—the complexity of the creative process is served well by total immersion in the activity ▪ <i>holistic view</i>—the full scope of a design problem is only fully embraced by taking a holistic, or systems, view. The designer needs to be able to take an overview position at any point and, in particular, to find multiple viewpoints of the data or emerging design important ▪ <i>parallel channels</i>—keeping a number of different approaches and viewpoints active at the same time is a necessary part of generating new ideas. <p>Exploration involves accessing source data that may be examined, assessed and interpreted in terms of the goals. This is an open process, possibly without observable directions, but, the thoroughness and selectivity of the activity is critical. Having a comprehensive set of knowledge sources readily available is extremely advantageous. Knowing where to look and how to select the knowledge is even more important.</p>	
<p><i>idea generation</i>—problem formulation, as distinct from problem solving, is critical to the effectiveness of the solution space that is defined. It draws upon a wide range of analogous cases often outside the immediate domain. This has been characterised as an ability to make remote associations. Creativity is demonstrated by the generation of many potential solutions instead of gravitating quickly toward a single and (usually) familiar solution that is not necessarily the optimal one. The ability to consider parallel lines of thought and to select and transform the results to meet the demands of a different situation is a critically important aspect of solution generation</p>	
<p><i>evaluation</i> - evaluation involves taking the results of the generative activity and testing the candidate solutions against a set of constraints. This leads to modifying, reformulating, or discarding solutions depending on the feedback. Selection of the optimal solution may involve a number of trade-offs against the constraints that are applied especially where, as is usually the case, the product is a complex one. The application of tight constraints may be considered conducive to creative solution finding and thus evaluation is a vital part of the creative process. Evaluation may be viewed as a pervasive activity that takes place from the exploration phase onward. The use of expert knowledge in evaluation has been identified as an important aspect of successful solution finding.</p>	

While acquiring specific domain knowledge is one of the course objectives, adaptiveness in generalising knowledge in order to enhance productive thinking as a basis for insight and true novelty of thinking is equally important. Productive thinking is the ability to use past experience on a general level, while still being able to deal with each new problem situation in its own terms. Gott, Hall, Pokorny, Dibble, & Glaser (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. The implication of this is effort spent on abstraction and reflection, well supported through the PBL process.

As previously noted, this learning environment is being implemented and evaluated during semester 1 2003. The results of this project will determine the future of PBL within the SE degree:

- if not successful, effort will need to be expended to ascertain why (since the literature suggests it should) and then to better construct the simulations and on staff development;
- if successful, to develop expertise in applying PBL within the domain. A follow-through will address the higher order learning outcomes that effect a change in student behaviour when tackling these problems.

Conclusions

Creativity has been described as a balance of convergent and divergent thinking appropriate to the situation (Nickerson, 1999). This balance is essential in undertaking software development, which may be considered as a class of creative problem solving. Providing a learning environment that enhances the opportunity for creative thinking has the potential to provide long term benefits to RE students, since there is evidence that students who have been taught to explore different ways to define problems (as in RE) engage in more creative problem solving over the longer term (Baer, 1988).

References

- Albert, R. S. (1996). Some reasons why childhood creativity often fails to make it past puberty into the real world. In M. A. Runco (Ed.), *Creativity from childhood through adulthood: new directions for child development* (pp. 43-56). San Fransisco: Jossey-Bass.
- Amabile, T. M. (1983). *The social psychology of creativity*. New York: Springer Verlag.
- Amabile, T. M. (1996). *Creativity in context*. Boulder (CO): Westview Press.
- Bach, J. (1999). Reframing requirements analysis. *IEEE Computer*, 32(2), 120-122.
- Baer, J. M. (1988). Long term effects of creativity training with middle-school students. *Journal of Adolescence*, 8, 183-193.
- Boud, D. (1985). Problem-based learning in perspective. In D. Boud (Ed.), *Problem-based learning in education for the professions* (pp. 13-18). Sydney: Higher Education Research Society of Australasia.
- Bubenko, J. (1995). *Challenges in Requirements Engineering: keynote address*. Paper presented at the RE'95: Second IEEE International Symposium on Requirements Engineering, York (UK).
- Carroll, J. M. & Swatman, P. A. (1999). *Opportunism in the Requirements Engineering process* (School of Management Information Systems Working Paper No. 1999/02): Deakin University, Australia.
- Edmonds, E. & Candy, L. (2002). Creativity, art practice and knowledge. *Communications of the ACM*, 45(10), 91-95.
- Gigch, J. P. V. (2000). Metamodelling and problem solving. *Journal of Applied Systems Studies*, 1(2), 327-336.
- Glass, R. L. (1995). A theory about software's practice (Editor's Corner). *Journal of Systems and Software*, 28, 187-188.

- Gott, S. P., Hall, E. P., Pokorny, R. A., Dibble, E. & Glaser, R. (1993). A naturalistic study of transfer: adaptive expertise in technical domains. In D. K. A. S. Detterman, (Ed.), *Transfer on trial: Intelligence, cognition and instruction* (pp. 258-288). Norwood (NJ): Ablex.
- Guindon, R. (1989). The process of knowledge discovery in system design. In G. Salvendy & M. J. Smith (Eds.), *Designing and using human-computer interfaces and knowledge based systems* (pp. 727-734). Amsterdam: Elsevier.
- Guindon, R. (1990). Knowledge exploited by experts during software systems design. *International Journal of Man-Machine Studies*, 33, 279-304.
- Koschmann, T. D., Myers, A. C., Barrows, H. S. & Feltovich, P. J. (1994). Using technology to assist in realising effective learning and instruction: a principled approach to the use of computers in collaborative learning. *The Journal of the Learning Sciences*, 3(3), 227-264.
- Lubars, M., Potts, C. & Richer, C. (1993). *A review of the state of the practice in requirements modelling*. Paper presented at the International Symposium on Requirements Engineering, San Diego.
- Maiden, N. & Gizikis, A. (2001). Where do requirements come from? *IEEE Software*, 18(5), 10-12.
- Maiden, N. A. M. & Sutcliffe, A. G. (1992). Exploiting reusable specifications through analogy. *Communications of the ACM*, 34(5), 55-64.
- Mann, J. (1996). *The Role of project escalation in explaining runaway information systems development projects: A Field Study*. Georgia State University.
- Nguyen, L. & Swatman, P. A. (2000). *Complementary Use of ad hoc and post hoc Design Rationale for Creating and Organising Process Knowledge*. Paper presented at the Proceedings of the Hawaii International Conference on System Sciences HICSS-33, Maui (Hawaii).
- Nickerson, R. S. (1999). Enhancing creativity. In R. E. Sternberg (Ed.), *Handbook of Creativity* (pp. 392-430). Cambridge (UK): Cambridge University Press.
- Patel, A., Kinshuk, & Russell, D. (2000). Intelligent tutoring tools for cognitive skill acquisition in life long learning. *Educational Technology & Society*, 3(1), 32-40.
- Pfleeger, S. L. (1999). Albert Einstein and empirical software engineering. *IEEE Computer*, 32(10), 32-37.
- Reeves, T. C. (1997). *A Model of the effective dimensions of interactive learning on the World Wide Web*. Paper presented at the Proceedings of Interaktiivinen Teknologia Koulutuksessa (ITK '97), Hameenlinna (Finland).
- Royce, W. W. (1970). *Managing the development of large software systems: concepts and techniques*. Paper presented at the IEEE WESCON.
- Standish. (1995). *Most programming projects are late*. West Yarmouth (MA): Standish Group.
- Sutcliffe, A. G. & Maiden, N. A. M. (1992). Analysing the novice analyst: cognitive models in software engineering. *International Journal of Man-Machine Studies*, 36, 719-740.
- Thomas, J., Lyon, D. & Miller, L. (1977). *Aids for problem solving* (No. RC-6468): IBM Research Report.
- Thomas, J. C., Lee, A., & Danis, C. (2002). Enhancing creative design via software tools. *Communications of the ACM*, 45(10), 112-115.
- Visser, W. (1992). Designers' activities examined at three levels: organisation strategies and problem-solving processes. *Knowledge-Based Systems*, 5(1), 92-104.
- Visser, W. & Hoc, J. (1990). Expert software design strategies. In J. M. Hoc, T. R. G. Green, S. R & G. D. J (Eds.), *Psychology of programming* (pp. 235-247). San Diego (CA): Academic Press.
- Wilson, B. G. & Cole, P. (1996). Cognitive teaching models. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (pp. 601-621). New York: Simon & Schuster Macmillan.
- Zimitat, C. & Alexander, H. (1999). The 4 Step Assessment Task (4SAT). *Integrity, Innovation and Integration*, 4, 692-697.