Using a Subject Area Model as a Learning Improvement Model

Errol Thompson
Department of Information Systems
Massey University
PO Box 756, Wellington 6000, New Zealand
E.L.Thompson@massey.ac.nz

Abstract

International literature suggests that to improve student learning in a subject area, strategies should be taught for learning the subject and how to assess the effectiveness of these learning strategies. In software development, the Capability Maturity Model® is used as process improvement model for software development. Can the model be adapted to provide a model for improving learning?

A model is proposed based on the principles of the Capability Maturity Model®. It is designed to encourage the learner to reflect on their learning and to evaluate the effectiveness of their learning.

Keywords: capability maturity, learning process, learning requirements, task representation, learning strategies, learning outcomes.

1 Introduction

Improved learning comes from a dual focus on the body of knowledge (content) and how to learn (process) (Murray-Harvey and Keeves, 1994). The process of learning how to learn involves the student in developing learning strategies, in the assessment of their learning, and in revising their strategies (Angelo and Cross, 1993, Bransford et al., 2000, Brown, 1978, Brown, 1987). From this understanding, is it possible to use a subject area model to develop a model for learning?

The Capability Maturity Model® is a tool for assessing the extent to which an organisation has established the processes to repeatedly develop high quality software to the customer's requirements on budget and on time (Chrissis et al., 2003). Using the model, an organisation is assessed and a path for improvement can be developed. After achieving the level five assessment of the model, the organisation will have in place the ongoing tools for self-assessment and improvement.

The Capability Maturity Model® (CMM) defines levels of maturity and a set of processes that should be in place if an organisation has developed its full potential for software development. Can the concepts of the CMM be transferred to the learning context? Is there a way of applying the principles that will enable the learner to identify the issues that they need to deal with in order to improve their learning? Would using the CMM in this way help reinforcing the principles of the CMM and its application in software development?

The model overviewed in this paper is a step in the process to developing a learning process maturity model (Thompson, 2004). The model can be applied continuously in a learning context to help the learner identify weaknesses and strengths in their approach to learning and in the selection of learning strategies.

2 What is a Learning process?

The idea of a learning process as a series of changes applies to the software developer as he/she moves from being a novice to being an expert. A novice software developer has a focus on detail while an expert will recognise and utilise patterns (Soloway et al., 1988, Pennington, 1987). Novices may seek example solutions while experts can develop solutions from concepts. Novices seek to apply design patterns and processes by rule while experts see these as guidelines to be adapted for the current task.

Within the ranks of novice programmers, this concept of a learning process as a series of changes also applies. There is a distinction between the person who struggles to learn programming (ineffective novice) and the person who can successfully learn to program (effective novice) with respect to the strategies that they utilise (Robins et al., 2003). The journey from ineffective novice to effective novice, and from effective novice to expert reflects the changing of perspectives in terms of what is being learnt and in the understanding of the task being performed (Mezirow, 2000, Mezirow, 1991, Kegan, 2000).

It is the viewing of process as a "series of stages in manufacture or some other operation" (Allen, 1990) that is used in the CMM. This process view gives us a way of being able to evaluate what a learner is doing or of being able to assist the learner to modify their approach to learning. Although not explicitly stated in all texts, this is the assumption that underlines the work of some authors when they describe steps that a learner can use (Harri-Augstein and Thomas, 1991, Senge et al., 1994).

In the field of software development, the OPEN process framework says, "The overall process itself consists of a number of Activities. Activities are granular descriptors of a collection of heterogeneous jobs that need to be done. These can be decomposed into a number of tasks,

each of which is readily ascertained as being either complete or in progress (i.e. incomplete). However, since Tasks have to be done by someone (or some thing such as another piece of software), it is useful to introduce a metaclass called Task performance, which consists of a Task and its producer." (Henderson-Sellers and Unhelkar, 2000).

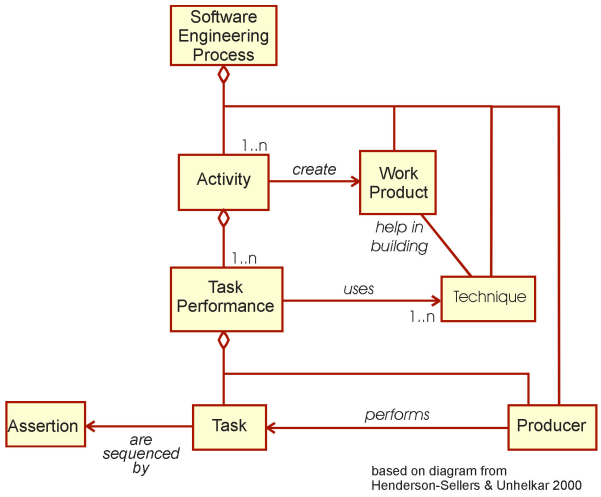

Figure 1: A Software Engineering Process

A software engineering process (see Figure 1) and the activities of a process are not aimless. They are intended to produce a tangible result (a work product or an artefact). In order to perform the activity and to produce the work product, a producer performs a task utilising a technique. Dependencies between tasks are defined by their assertions. An assertion includes pre-conditions that define what is required for the task to start and post-conditions that define what the task has produced. The workflow represents the sequence of tasks performed.

Based on this perspective of process, we would expect a learning process to involve a series of activities each aimed at achieving some aspect of the learning (a work product). We would expect that the work product would be assessable. That assessment of the work product should reflect the degree of learning that has occurred but the assessment of the work product does not reflect the effectiveness of the process used to obtain that outcome.

When reflecting on the activities and work products of software development, we see that these can include the clarification of objectives (requirements), identification, selection, and development of appropriate solution strategies (analysis and design), determining a method for testing the solution (writing tests), preparing the solution (coding), assessment of the solution (testing), reviewing progress and reassessing further requirements (review and planning the next iteration), and transition to production (implementation) (Firesmith and Henderson-Sellers, 2002, Henderson-Sellers and Unhelkar, 2000, Whitten et al., 2001, Valacich et al., 2001, Chrissis et al., 2003, Paulk et al., 1993). The Open Process Framework (Henderson-Sellers and Unhelkar, 2000, Firesmith and Henderson-Sellers, 2002) defines a high level process model and provides a menu of activities and tasks from which specific processes can be built.

With respect to learning, the activities and work products should also cover determination or clarification of learning requirements (learning outcome determination), identifying ways of obtaining the required learning outcomes (building a series of task representations and selecting learning strategies), defining how to determine whether the learning has been achieved (define the learning outcome criteria), carrying out the learning (using the selected learning strategies), determining whether the learning has been achieved (assessment), identifying further learning needs, and applying learning in a work or non-learning context (application of learning). Like the OPEN process framework, we would expect that it would be possible to list activities and tasks that could be selected from to build a specific learning process.

3 What is maturity?

In software development, the maturity of an organisation's capability to develop software may be defined as the organisations ability to "repeatedly and reliably deliver customers requests" (Poppendieck, 2003) or the extent to which an organisation that has established the processes to repeatedly develop high quality software to the customer's requirements on budget and on time (Chrissis et al., 2003). Maturity in the CMM relates to the organisation and its institutionalization of the processes (Chrissis et al., 2003).

In an educational context, learning maturity may be defined as the extent to which a person has developed their capability to repeatedly and reliably achieve learning outcomes that involve ability to apply, critique, analyse, reflect, and hypothesise on the subject under study. A mature learner will have fully developed powers of learning where powers of learning may be defined in terms of the cognitive and metacognitive skills (Anderson et al., 2001, Bloom et al., 1956, Facione, 1990, Flavell, 1979, Brown, 1978, Brown, 1987) that characterise deep and critically reflective approaches to learning (Biggs, 1999, Biggs and Collis, 1982, Schön, 1983, Mezirow, 2000). The mature learner accepts changes to their perspective of learning and of the subject matter (Mezirow, 1991) and commits to their current understanding based on sound reasoning (King and Kitchner, 1994, Polanyi, 1958, Perry, 1968) and the processes of the subject area (Costa and Liebmann, 1996).

Identifying whether a learner is using deep, achieving or surface strategies is inadequate to determine learning maturity (Biggs and Collis, 1982) since the selection of a learning strategy relates to the learner's task representation rather than to a characteristic of the learner (Hunt, 1995, Ramsden, 2003). Even in a task representation that may involve the characteristics of a deep approach to learning, the learner may utilise surface strategies to build a knowledge base before endeavouring to utilise the strategies of deep approaches to learning. Webb (1997a, 1997b) argues that a learner during the learning process may switch dynamically between surface and deep approaches to learning depending on their need to develop an understanding of the terminology required.

There is sequencing in the use of strategies depending on the learner's prior knowledge and the learning task at hand. A mature learner is able to select appropriate strategies based on their prior knowledge and learning task representation.

Prior knowledge has been identified as a key factor in learning success (Hunt, 1995). Bransford et al. (2000) emphasise the need to "draw out and work with the pre-existing understandings" that students bring with them (p 19). In the context of a process model for learning, we would expect to see that a mature learner's learning process works to identify learning strategies that will identify relevant pre-existing understandings and knowledge, and to uncover connections between the current learning and these prior understandings and knowledge.

4 The Capability Maturity Model

For the purposes of developing a learning maturity model, our primary interest is the principles of the maturity levels. In this discussion, the levels of the staged representation of the Capability Maturity Model® are utilised (CMMI Product Team, 2002). Each level of the Capability Maturity Model® sees a change of perspective with respect to the process. At level 1, the focus is on the tasks that need to be done to achieve the required outcome. The requirements must be identified and defined. Monitoring and managing the process are not considered. At level 2, the process is seen as something that can be planned and managed. There is a sense in which the process is becoming a rule to be followed for success. In order to define a process, there is a need to understand the path that is to be taken and to manage changes that might occur through changed requirements. At level 3, there is recognition that a standard process can be defined, applied, and be tailored for a specific task. The selected process model may be followed more by rule with tailoring being the inclusion or exclusion of steps based on whether the desired outcome is required. At level 4, there is a move to assessing the effectiveness of the process in achieving the desired quality of result. The process is open to evaluation and critique. At level 5, the process is seen as continually open to change and improvement. The process moves to being a guide and not a rule. A changed process can be used but it is used in a context of being evaluated.

5 Learning Process Maturity Model

With its focus on assessing organisational process maturity, some of the processes and levels do not seem to map easily to the individual. Humphrey (1995, 1997) has developed, in the personal software process (PSP), a strategy where the individual software engineer can apply the principles of the capability maturity model to their own work. Although Humphrey makes no attempt to redefine the maturity levels for the individual, he does identify the process areas that are relevant to the individual and develops a series of steps for implementing personal process improvement. He says that the role of the PSP is "to understand the CMM and know how to apply its principles" (Humphrey, 1995).

The following learning maturity levels are based on the Capability Maturity Model® (Chrissis et al., 2003). The focus on process is primarily as a series of activities rather than stages of development.

5.1 Learning maturity level 1: Initial or immature

At level 1, initial or immature, the learner relies on direction from the learning facilitator (i.e. is a dependent learner) or grabs at random from a bundle of known learning strategies. The success of any learning venture is either dependant on the guidance given or on the accidental selection of an appropriate strategy. There is little structure to how learning is carried out. An immature learner may focus almost exclusively on memorizing facts or seeking right solutions (dualistic right or wrong thinking (Perry, 1968)) and may not recognise the thinking processes or strategies of the subject area. In seeking advice or assistance, this type of learner will be seeking clarification of whether they are right. They may resist efforts to make them think about the reasoning for a particular outcome. Most often the learner will see learning as occurring only in formal learning situations. This doesn't mean that they are not learning outside these situations. Rather their perception of learning is that it occurs in a formal setting.

5.2 Learning maturity level 2: Managed

At level 2, managed, the learner has identified a number of strategies that have been successful in obtaining the desired learning outcomes. At this stage, the strategies are not clearly defined but they are things that the learner has become accustomed to doing when endeavouring to learn and the strategies being used have delivered success on a reasonable number of occasions. The learner is tracking how they allocate their time to the learning strategies and planing how they will approach their learning.

The tracking of how time is spent doesn't reflect the effectiveness of how that time is spent. The identification of where time is spent may allow decisions to be made with respect to time usage. The learner might drop activities that seem unrelated to the learning in favour of spending more time on activities that seem to relate to learning.

As trial of using time recording, I began to record how I used my time as I worked on developing this model. I was spending a large amount of time on activities related to developing this model. That is I was focussing on time from a *chronos* perspective. I was regarding all time as of equal value and was concerned about the amount of time spent on task (Covey et al., 1994). However, I didn't seem to be making much progress and I wasn't seeing much in the way of outputs. Changing my strategy to focus on velocity (the tasks that I completed in a time period (Beck, 2000)), saw my productivity increase. My focus switched to a *kairos* view of time where the focus is on the quality of time usage (Covey et al., 1994).

Both methods provided valuable information for my time management. The first method told me where I was spending my time. I quickly identified that the bulk of my

time was spent on research focussed activities. Time recording didn't require tasks with identifiable completions. I could time record based on activities performed. The second method helped provide a focus on results rather than on activities. The second method saw me focus on tasks with definable end points. An activity might be to read research papers. A task is to read a specific research paper and to summarise the key points. I can tell when the task is complete.

Focussing on tasks and their completion also meant that there was a need to more clearly define what was required. I had to define what completion of a task meant and the criteria that would define completion. I had to understand the requirements of the task before I began the task. This type of thinking is reflected in test-driven development where an automated test is written and then the code that satisfies the test (Beck, 2003, Astels, 2003).

A second element of learning maturity level 2 is a focus on understanding what is required and not simply what needs to be done. In a course of study, a learner may be assigned a reading. What will determine whether the time spent completing that reading was effective? The criteria for a learning task have to reflect more than the activity. They have to define the standard of performance. The standard of performance may be defined using a cognitive skills hierarchy (Anderson et al., 2001).

At this maturity level, the learner possibly with the assistance of their learning facilitator is identifying their current knowledge, the desired learning outcomes, learning strategies to use, and required resources. They are able to plan an approach to achieving the desired learning outcome. The quality of the learning is not being assessed rather the effectiveness of the plan is based on a perception of success against the defined learning outcomes.

5.3 Learning maturity level 3: Defined

At level 3, defined, the learner has a number of strategies that they use on a regular basis. They are defined in the sense that the learner sees these strategies as being used consistently within the subject area. The learning strategies may be linked with some model of learning or learning preferences.

The learner is recognising that the subject area solution possibilities are drawn from processes that reveal them and are supported by subject area reasoning (Costa and Liebmann, 1996). Knowledge may be seen more in terms of a current state of commitment based on supporting evidence rather than being absolutely "correct" (Polanyi, 1958). The learner may seek reassurance from an authority for the reasoning and supporting evidence for knowledge (Perry, 1968). The learner may still have some dependence on the learning facilitator for the processes, subject area reasoning, and assessment of learning.

At this level, the learner is looking beyond understanding of the end requirements and is beginning to explore and understand the required learning journey. During this process, a series of transitions may be required. Completion of a task may not be enough. There may be a need to develop additional background knowledge in order to be able to complete a learning task as defined by a learning facilitator.

This is like a software developer who is asked to produce a web services application. The programmer considers their current knowledge and status of the applications that already exist. They decide what they need to learn and exactly what work needs to be done before producing any code. They don't want to repeat work. They may have had years of experience writing applications using object-oriented approaches for local users but no experience with web applications or web services. The programmer doesn't start by attempting to write the application. Rather they set out to learn about web services and what is needed to write such an application. Once they have this knowledge in place, they are ready to start the development of the application. Because it is a new environment, they may call on the services of someone with experience of writing such applications to evaluate and check their approach. The programmer is recognising the journey that is involved in developing the application and not just the task that seemed to be originally stated.

A learner also needs to take the learning journey into account. They can not sign up for an advanced course in a subject without having learnt the prerequisite material to the required standard. As learning facilitators, we recognise this in planning programmes of study. However, at a micro level a learner needs to recognise the journey that they need to take from their current knowledge levels to the knowledge levels required to complete the learning task.

5.4 Learning maturity level 4: Assessed

At level 4, assessed, the learner is beginning to be able to measure the effectiveness of their learning. At this level, they are beginning to see that their known strategies are not always the most efficient and are seeing how to assess their strategies for different learning tasks. Because they are beginning to measure the effectiveness, they are also more open to new strategies. The learner is developing a degree of independence in their learning situations.

In previous levels, the criteria for completion of a learning task have primarily come from external sources. Here, the learner is beginning to develop the skills to define their own criteria and to some extent their own learning tasks. In this respect, the learner becomes less dependent on formal learning situations and life long learning becomes more likely.

Relating this to software development, it is like a developer who may have depended on professional software testers in order to determine whether they have achieved the requirements for the code that they have completed. Now, the programmer may be able to define a range of automated tests that cover the requirements. These tests may be drawn directly from the user requirements and relate closely to the customer acceptance tests. When they pass their code on to quality assurance or the customer, they have a higher level of confidence that the code is satisfying the requirements. This doesn't mean total independence from the customer

in defining requirements but rather the programmer now endeavours to ensure that they understand the requirements and how they can assess that those requirements are satisfied.

This approach to programming may also be giving the developer a clearer understanding of progress in terms of the number of customer requirements satisfied during a period of development activity. The developer is more aware of the effectiveness of their practices and may have in place additional measures that signal the difference between effective and ineffective development processes. Quality of performance is of equal importance as productivity. Practices are being implemented to ensure quality as well as progress.

At this level, the learner is open to change with respect to how they perceive learning and the outcomes of learning. They become more focussed on building conceptual frameworks and developing the skills for thinking and reasoning in the subject area. The quality of learning is perceived not simply by how much has been learnt but by the ability to reason and evaluate the thinking processes used in the subject area. At this level of maturity, the learner is beginning to assess their outcomes of their learning with a high degree of accuracy.

5.5 Learning maturity level 5: Optimizing

At level 5, optimizing, the learner has become quite flexible in the use of learning strategies. When a learning situation presents itself, the learner evaluates strategies and selects, from their repertoire, a strategy that is the most appropriate for the given circumstances. The learner has developed a clear understanding of the effectiveness and appropriateness of each learning option they have available. They are also keen to expand their repertoire so that they can face new and more challenging learning situations. There is a desire for ongoing improvement of their learning and understanding.

This is like the developer who instead of following a specific methodological approach is able to select an approach that matches the current development task. Even when they select a methodological approach, they do not ignore techniques from other approaches but select those that will be of benefit to the current work. This selection is done in an environment of on going evaluation of the effectiveness of what they are doing. If an approach is found to be less effective, it may be discarded in favour of another approach that, based on measurement, is delivering more favourable results.

At this level, the learner is not dependent on specific learning situations or on direction. They see learning possibilities all around them and take on the challenges that are available. Conceptual frameworks, thinking skills and reasoning in the subject area are the primary focus. So-called facts, solutions, processes, and reasoning are not taken at face value. Everything is open to evaluation and interpretation. The learner seeks to commit to their current understanding and thinking processes based on sound supporting reasoning (Perry, 1968).

When problems are identified with a learning approach, the learner seeks to understand and address the cause. Like a project leader, the learner endeavours to identify bottlenecks and constraints to their learning, thus is better equipped to resolve them.

6 Proposed usage

The next stage in this research is to develop a set of tools that can be used within the normal teaching cycle for undergraduate first and second year programming courses or papers. The tools will be designed to assist the learner to move through the maturity levels.

In effect this would be the development of a personal learning process that has some similarities with the personal software process.

The first set of tools will be aimed at moving the learner from learning maturity level 1 (initial or immature) to learning maturity level 2 (managed). These tools will be diagnostic in the sense of helping the learner to discover their current practices and to plan improvement. The tools will involve time recording or progress recording, learning strategy identification, project planning, assessing current knowledge levels, and defining learning requirements. Some of these tools will draw upon those already used in project management or as classroom assessment techniques (Angelo and Cross, 1993).

The second set of tools will be aimed at moving the learner from learning maturity level 2 (managed) to learning maturity level 3 (defined). The tools will involve presenting possible learning process strategies and the contexts in which they apply.

The third set of tools will be aimed at moving the learner from learning maturity level 3 (defined) to learning maturity level 4 (assessed). The tools will involve assessment of the quality of learning achieved by using a learning assessment strategy (Biggs, 1999, Biggs and Collis, 1982) and where the learner is with respect to models of learning (Perry, 1968, Mezirow, 2000, Mezirow, 1991, King and Kitchner, 1994).

The fourth set of tools will be aimed at moving the learner from learning maturity level 4 (assessed) to learning maturity level 5 (optimizing). The tools will involve being able to identify weaknesses or defects in current learning strategies and to identify new strategies that will address those weaknesses and reduce the defects. This may draw on principles of effectiveness (Covey, 1990, Covey et al., 1994) and on process improvement strategies particularly those that focus on constraints and elimination of constraints (Goldratt, 1990, Goldratt and Cox, 1993, Poppendieck and Poppendieck, 2003).

7 Conclusion

The model proposed forms the basis for the development of a learning process maturity model. It is process focussed in that it concentrates on the planning and managing the strategies of learning rather than the stages of development. It is recognised that as a learner moves through the stages, there will be a change in attitude with

respect to the objectives of learning and as a consequence the assumptions about the use of learning strategies.

The model is a framework for thinking about learning problems. Since the model uses concepts from software development maturity model, it provides a framework to improve learning and to improve strategies for software development.

This model is subject area specific but can it be applied to other subjects? There may be some application outside of software development but that is not the argument here. The concepts of process and maturity may be unfamiliar to other subject areas. If anything, the recommendation of this model is that other subjects should explore their own domain to integrate learning theory into the concepts of that domain.

8 References

Allen, R. E. (1990) *The concise Oxford dictionary of current English,* Oxford, Claredon Press.

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J. & Wittrock, M. C. (Eds.) (2001) *A taxonomy for learning and teaching and assessing: A revision of Bloom's taxonomy of educational objectives*, Addison Wesley Longman.

Angelo, T. A. & Cross, K. P. (1993) *Classroom assessment techniques: A handbook for college teachers,* San Francisco, Jossey-Bass Publishers.

Astels, D. (2003) *Test-driven development: A practical guide,* Upper Saddle River, Prentice Hall PTR.

Beck, K. (2000) *Extreme programming explained: Embrace change,* Boston, Addison Wesley Longman.

Beck, K. (2003) *Test-driven development by example,* Boston, MA, Addison Wesley.

Biggs, J. B. (1999) *Teaching for quality learning at University,* Buckingham, Open University Press.

Biggs, J. B. & Collis, K. F. (1982) *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome),* New York, Academic Press.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H. & Krathwohl, D. R. (1956) *Taxonomy of educational objectives Handbook 1: cognitive domain,* London, Longman Group Ltd.

Bransford, J. D., Brown, A. L. & Cocking, R. R. (Eds.) (2000) *How people learn: brain, mind, experience, and school,* Washington, National Academy Press.

Brown, A. L. (1978) Knowing when, where, and how to remember: A problem of metacognition. In Glaser, R. (Ed.) *Advances in instructional psychology.* Hillsdale, NJ, Lawrence Erlbaum Associates.

Brown, A. L. (1987) Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In Weinert, F. E. & Kluwe, R. H. (Eds.) *Metacognition, motivation, and understanding.* Hillsdale, NJ, Lawrence Erlbaum Associates.

Chrissis, M. B., Kourad, M. & Shrum, S. (2003) *CMMI Guidelines for process integration and product improvement,* Boston, Addison Wesley.

CMMI Product Team (2002) CMMI[SM] for systems engineering/software engineering/integrated product and process development/supplier sourcing, version 1.1 (CMMI-SE/SW/IPPD/SS, V1.1) staged representation. Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University.

Costa, A. L. & Liebmann, R. M. (Eds.) (1996) *Envisioning process as content: Toward a renaissance curriculum,* Thousand Oaks, CA, Corwin Press Inc.

Covey, S. R. (1990) *The seven habits of highly effective people,* New York, Simon and Schuster.

Covey, S. R., Merrill, A. R. & Merrill, R. R. (1994) *First things first: to live, to love, to learn, to leave a legacy,* New York, Simon and Schuster.

Facione, P. A. (1990) Critical thinking; A statement of expert consensus for purposes of educational assessment and instruction, research findings and recommendations. Fullerton.

Firesmith, D. G. & Henderson-Sellers, B. (2002) *The OPEN process framework: An introduction,* London, Addison Wesley.

Flavell, J. H. (1979) Metacognition and cognitive monitoring: A new area of cognitive-development inquiry. *American Psychologist,* 34**,** 906-911.

Goldratt, E. M. (1990) *Theory of constraints: What is this thing called Theory of Constraints and how should it be implemented?* New York, North River Press.

Goldratt, E. M. & Cox, J. (1993) *The goal: a process of ongoing improvement,* Aldershot, Gower Publishing.

Harri-Augstein, S. & Thomas, L. (1991) *Learning conversations: The self-organised learning way to personal and organisational growth,* London and New York, Routledge.

Henderson-Sellers, B. & Unhelkar, B. (2000) *OPEN modeling with UML*, Addison Wesley.

Humphrey, W. S. (1995) *A discipline for software engineering,* Reading, MA, Addison Wesley Longman.

Humphrey, W. S. (1997) *Introduction to the personal software process [(tm)],* Reading, MA, Addison Wesley Longman.

Hunt, L. M. (1995) Approaches to learning: The selection and use of learning strategies. Palmerston North, Massey University.

Kegan, R. (2000) What "form" transforms? A constructive-developmental approach to transformative learning. In Mezirow, J. & Associates (Eds.) *Learning as transformation: critical perspectives on a theory in progress.* 1st ed. San Francisco, Jossey-Bass.

King, P. M. & Kitchner, K. S. (1994) *Developing reflective judgement: understanding and promoting intellectual growth and critical thinking in adolescents and adults,* San Francisco, Jossey-Bass Inc.

Mezirow, J. (1991) *Transformative dimensions of adult learning,* San Francisco, Jossey-Bass Publishers.

Mezirow, J. (2000) Learning to think like an adult: Core concepts of transformation theory. In Mezirow, J. & Associates (Eds.) *Learning as transformation: critical perspectives on a theory in progress.* 1st ed. San Francisco, Jossey-Bass.

Murray-Harvey, R. & Keeves, J. P. (1994) Student's learning processes and progress in higher education. New Orleans, LA, American Educational Research Association.

Paulk, M. C., Curtis, B., Chrissis, M. B. & Weber, C. V. (1993) Capability maturity model[SM] for software, version 1.1. *Capability maturity modelling.* Pittsburgh, PA, Carnegie Mellon University.

Pennington, N. (1987) Comprehension strategies in programming. In Olson, G. M., Sheppard, S. & Soloway, E. (Eds.) *Empirical studies of programmers: Second Workshop.* Norwood, NJ, Ablex Publishing.

Perry, W. G., Jr. (1968) *Forms of intellectual and ethical development in the college years: A scheme,* Fort Worth, Harcourt Brace Jovanovich College Publishers.

Polanyi, M. (1958) *Personal knowledge: towards a post-critical philosophy,* Chicago, Routledge and Kegan Paul Ltd.

Poppendieck, M. (2003) Toward a new definition of maturity.

Poppendieck, M. & Poppendieck, T. (2003) *Lean software development: An agile toolkit,* Boston, Addison Wesley.

Ramsden, P. (2003) *Learning to teach in higher education,* London, Routledge Falmer.

Robins, A., Rountree, J. & Rountree, N. (2003) Learning and teaching programming: A review and discussion. *Computer Science Education,* 13**,** 137-172.

Schön, D. A. (1983) *The reflective practitioner: how professionals think in action,* New York, Basic Books.

Senge, P., Kleiner, A., Roberts, C., Ross, R. & Smith, B. (1994) *The fifth discipline fieldbook, strategies and tools for building a learning organization,* London, Nicholas Brealey Publishing.

Soloway, E., Adelson, B. & Ehrlich, K. (1988) Knowledge and processes in the comprehension of computer programs. In Chi, M. T. H., Glaser, R. & Farr, M. J. (Eds.) *The nature of expertise.* Hillsdale, NJ, Lawrence Erlbaum Associates.

Thompson, E. (2004) Towards a learning process maturity model. In Freyberg, C., Hartmann, S., Kaschek, R., Kinshuk, Schewe, K.-D. & Turull Torres, J. M. (Eds.) *PhD Workshop 2004.* Palmerston North, Department of Information Systems, Massey University.

Valacich, J. S., George, J. F. & Hoffer, J. A. (2001) *Essentials of systems analysis and design,* Upper Saddle River, NJ, Prentice Hall.

Webb, G. (1997a) Contesting learning theory: a response to Entwistle and Ekeblad. *Higher Education,* 33**,** 225-230.

Webb, G. (1997b) Deconstructing deep and surface: Towards a critique of phenomenography. *Higher Education,* 33**,** 195-212.

Whitten, J. L., Bentley, L. D. & Dittman, K. C. (2001) *Systems analysis and design methods,* Boston, MA, Richard D Irwin, McGraw-Hill.