



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

Box, I. (2003) Using formative assessment and the feedback process as an approach to the learning of software development review and quality assurance skills, in *Learning for an Unknown Future, Proceedings of the 26th HERDSA Annual Conference, Christchurch, New Zealand, 6-9 July 2003: pp 61.*

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.

Using formative assessment and the feedback process as an approach to the learning of software development review and quality assurance skills

Ilona Box

University of Western Sydney, Penrith, Australia
i.box@uws.edu.au

Abstract: *There are many factors that can contribute to improving the learning experiences of students. Those that are of interest here include: formative assessment and the process of providing feedback and its similarity to the software development review process. Based on Scriven's original definition of formative evaluation involving a product, a developmental agency and a feedback loop, four feedback loops are suggested: judicious teacher/responsive student; advisory teacher/judicious student; judicious peer/responsive student; and self-determining student. The use of the feedback loops revealed some strengths and weaknesses of each.*

Keywords: *feedback, formative assessment, software quality assurance*

Introduction

There are many factors that can contribute to improving the learning experiences of students. Those that are of interest here include: formative assessment and the process of providing feedback.

In my discipline, software development, students with a deep approach to learning (e.g., Entwistle, 1987) take on the responsibility to participate in the achievement of learning objectives without necessarily receiving an immediate (i.e., during the teaching session) extrinsic reward (e.g., a high mark). They are also able to realistically evaluate the outcomes of their learning i.e., software development artefacts (henceforth referred to as artefacts) and their skill to develop these. On completion, the students with a deep learning approach present with a professional, though novice, skill level for doing the tasks and the quality assurance process associated with the tasks. Students that take a surface approach to learning are not connecting their learning with the achievement of learning objectives and are unable to realistically evaluate their work. Students who expect immediate and accumulating extrinsic rewards (e.g., full marks for tasks allocated a few percentage points which when added together will allow them to pass the subject) by producing artefacts take a strategic approach to learning. Though the students with a strategic approach to learning (Entwistle, 1987) may be able to evaluate the artefacts, they do not see the production of the artefacts as valuable to their learning or their professional career. The most desirable approach to learning software

development is a deep approach. A contribution can be made to encourage the development of a deep approach to learning software development by appropriate course design and execution, one element of which is the feedback process related to formative assessment.

Two major laws of skill acquisition need to be considered when using formative assessment these are the "power law of practice" and "knowledge of results" (Pelligrino, Chudowsky, & Glaser, 2001, p. 85). Research of assessment, though regarded as uneven and weak (Black, 2000) presents a consensus that learning objectives are achieved by the use of reliable and valid assessment (e.g., Biggs, 1999; Boyd, 2001; Chase, 1999; Crooks, 1994; Kanjee, 2000). Using such assessment and appropriate feedback processes therefore yields positive learning outcomes for the students. However, if we encourage a deep learning approach there needs to be an opportunity for the students to practice doing the assessment and responding to feedback.

In 1931 Thorndike demonstrated that practice without feedback produces little learning (Pelligrino et al., 2001). A student needs to have knowledge of their mistakes. This is provided by feedback to students during instruction and learning. The requirements for effective feedback are (Crooks, 1994, p. 12):

1. "... truly informative, clearly identifying ... strength[s] and weakness[es] of the individual student."
2. "ways of improving ... should be ... explicit as possible."
3. "delivered ... to enhance rather than damage student motivation."
4. "timely."

To provide an environment that effectively facilitates student practice doing the task and responding to feedback, course designers need to consider the process by which feedback will occur. This paper presents four feedback processes or loops. The feedback loops have been used and this use has indicated the some strengths and weaknesses of each.

Formative assessment

Generally, assessment can be described as either summative or formative. Summative assessment takes place at the end of a unit of study and "sums up" the current level of knowledge and skill a student will have of that unit of study.

Formative evaluation was first used by Scriven (1973) as the term to describe the evaluation of a curriculum while it is still under development. Formative evaluation has since been widely adopted as the term to describe "systematic evaluation in the process of curriculum construction, teaching, and learning for the purpose of improving any of these three processes" (Bloom, Hastings, & Madaus, 1971, p. 117). Bloom et al. went further by describing the use of formative evaluation to improve the teaching and learning processes. Formative evaluation, in the Australian context, is the same as formative assessment.

Formative assessment is used to determine whether a learner has achieved an adequate level of skill or mastery of some subject content before the opportunity to learn that subject content has passed. Formative assessment involves providing feedback and an opportunity for improvement to take place. Providing feedback requires a judgement about the quality of the work, which can be explicitly indicated by comments and sometimes with an indication of the level of mastery such as a temporary grade. Providing feedback is similar to the review and

quality assurance process during software development (space does not permit a discussion of this process, however, one can be found in Box & Ferguson, 2002).

For most of us the time available to deliver a course is quite rigid, typically restricted to an institution's dateline which includes not only academic, but also administrative deadlines. Therefore the opportunity to learn the subject content is finite, which influences the amount of formative assessment and feedback that can take place. This is similar to a project plan where the time available for quality assurance is also finite.

Formative assessment is interpreted in a number of ways. Some (for instance, Black & William, 1998; Bloom et al., 1971; Cowie & Bell, 1999; Torrance, 2000) consider the assessment task to be formative when the judgment of the student's work is made by the teacher. A series of formative assessments, of this type, can be regarded as "a multiple set of summative assessments" (Bligh, 2000), which is not analogous with the software development review process. Formative assessment does require some judgment to be made of the quality of the student's attempt at the assessment task and effort is needed to reduce the negative aspects so the formative assessment is perceived as an opportunity for improvement just as a review is used to prevent errors in the software. Reducing the negative aspects associated with judging is in line with the view that formative assessment should be non-judgemental and non-threatening (Rolfe & McPherson, 1995). Bloom et al. (1971, p. 54) note formative assessment is "part of the learning process, and should in no way be confused with the judgement of the student's capabilities or included in the grading process". In the same way a review of an artefact is not a personal criticism of the software developer. Therefore, the feedback provided to the software development student is about what is 'right' about the artefact and what and how it needs to be improved. The judgement being made is about students' progress in their learning, not whether they have or have not achieved or are capable of achieving a particular grade.

Given that 1) we work to rigid datelines, 2) software development students may not initially have the skill and subject expert knowledge to realistically evaluate their work, 3) evaluating one's own and others work is integral to being a successful software developer, then it is necessary for the students' learning software development to experience formative assessment where there is a move from the teacher being the reviewer of the artefacts, to where the student is the reviewer. Providing feedback processes that are clearly defined is a way this move can be achieved.

Feedback

Scriven first used formative evaluation in 1967 as the term to describe the evaluation of curriculum while it is still under development. Formative evaluation has since been widely adopted for student assessment. By returning to Scriven's work on formative evaluation I was able to define frameworks, feedback loops, by which students may learn how to provide and use feedback, similar to the software development review process.

Scriven (1973, p. 63) states that during formative assessment "the evaluation feedback loop stays within the developmental agency (its consultants), and [that it] serves to improve the product". To establish what would constitute formative assessment and how we might move the onus of quality assurance on to the student I needed to decide what is the product, what is the developmental agency, and what is the feedback loop.

The product

Minimally and in the short term, the primary product is student learning. Within a single subject this is usually demonstrated by the completion of one or more summative assessment tasks. For a smaller unit of study the level of mastery can be demonstrated by the use of formative assessment. It is most desirable for all students to achieve a minimum level of mastery. However, a rigid dateline makes this virtually impossible to achieve unless there is some sacrifice of content and/or standards, and/or a guarantee of homogeneity of the prior knowledge of all students, the former being more achievable than the latter and the sacrifice of content being more palatable than the sacrifice of standards.

Elaborately and in the longer term, the primary product is a professional software developer. Except when a student continues into a research degree, it is not necessarily possible for us to know if the student has become a professional software developer.

The developmental agency

In Scriven's definition the developmental agency is usually seen as academics and external to where the curriculum is in place. Developing is the bringing out of capabilities and an agent is one who acts or has the power to act (Delbridge, 1997). Based on these definitions I defined the developmental agency for my circumstances as internal to the delivery of the subject and as either the teacher; the teacher and student; or the student.

Before formative assessment commences the teacher is the developmental agency. Once a student has moved away from the teacher's domain, the teacher is again the developmental agency. However, during formative assessment it is either the teacher and student, or the student who comprise[s] the developmental agency. Formative assessment is then a simulation of the software development process where, the teacher is a reviewer and the student is a developer or, the student is a reviewer and developer.

The feedback loop

Based on Scriven's work, as it aligned nicely with the software development process, I formulated and used a number of feedback loops aimed at students improving the quality of their software development work and their skill at reviewing and assuring the quality of their own and others artefacts, namely:

1. Judicious teacher/responsive student;
2. Advisory teacher/judicious student;
3. Judicious peer/responsive student;
4. Self-determining student;

'Judicious' indicates that the role of the person is to exercise good judgement about the quality of the artefacts, expressed as feedback that is wise, sensible, and assists learning. 'Responsive' indicates that the role of the person is to respond to feedback provided by improving the quality of the artefact. 'Advisory' indicates that the role of the person is to facilitate another's judicious role. 'Self-determining' indicates that the role of the person is to be judicious and responsive.

The judicious teacher/responsive student feedback loop

The first feedback loop—judicious teacher/responsive student—is shown in Figure 1. The model uses the Unified Modelling Language (a software development standard) sequence diagram notation. The vertical bar represents the 'life' of the assessment activity, time progresses as one moves down the vertical bar. The teacher commences the activity by setting

the task and criteria against which the task may be judged. The student responds with an attempt at the task, using the criteria as a guide to achieve mastery. The teacher provides comments and an indicative grade of the level of mastery achieved on the attempt at the task. The student responds with another attempt at the task. This loop can continue until the student demonstrates an adequate mastery of the task or as long as time permits (which may mean in a group of students that some do not achieve an adequate mastery of the task). The judicious teacher/responsive student feedback loop leaves the onus of making good judgments about the quality of the artefact on the person with the most skill and subject expert knowledge, the teacher.

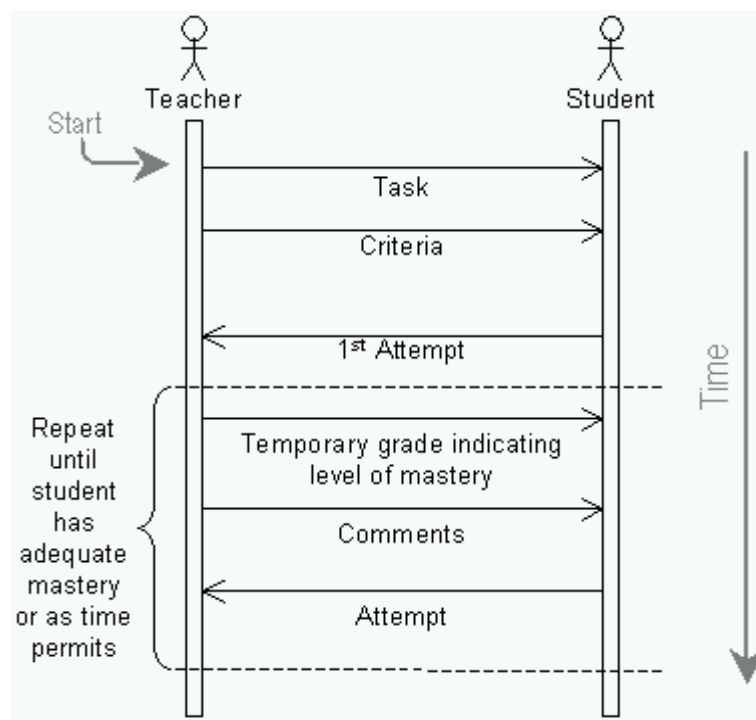


Figure 1: The judicious teacher/responsive student feedback loop

I used the judicious teacher/responsive student feedback loop when students first learned about software development, and the review and quality assurance process with some success (Box, 2002).

From my experience with the judicious teacher/responsive student feedback loop, some of the strengths of the loop are:

- The judgments are more likely to be correct (meeting some of Crooks' (1994, p. 12) effective feedback requirements, i.e., truly informative, clearly identifying strengths and weaknesses, ways of improving are explicit, and delivered to enhance motivation) as the teacher possesses the subject expert knowledge;
- The time to achieve adequate mastery, for the majority of the class, is shortened because the teacher can steer students to the right course of action;
- The student can concentrate on responding to comments and doing the task.

Some of the weaknesses of the judicious teacher/responsive student loop I found to be:

- The student is not active in making judgments and is less likely to acquire this skill by only being an observer of the judgment process and a recipient of the outcome;

- The workload of the teacher might increase and becomes more time-critical (to meet Crooks' effective feedback requirement of being timely).
- Prolonged use of this feedback loop is less likely to result in software developers who can independently review and assure the quality of their work and others.

The weaknesses of the judicious teacher/responsive student feedback loop were addressed in the formulation of the next loop.

The advisory teacher/judicious student feedback loop

The second feedback loop, advisory teacher/judicious student (Figure 2), has the student in a more active role of making judgments about their attempts at work set by the teacher. The teacher plays a role of setting the task and criteria against which the task may be judged. The student attempts the task and, using the criteria, comments on the attempt at the task. The student can attempt and judge the task as often as they like, or as time permits. The length of the vertical bar indicates the “life” of each attempt and comment. The teacher can advise the student as much and as often as the student wants. The teacher's advice can be directed more at the judgments the student is making, that is to say the student's interpretation of the criteria and how they apply to their attempt at the task, than the attempt itself. The loop would end when the task is treated as a summative assessment and the teacher grades it.

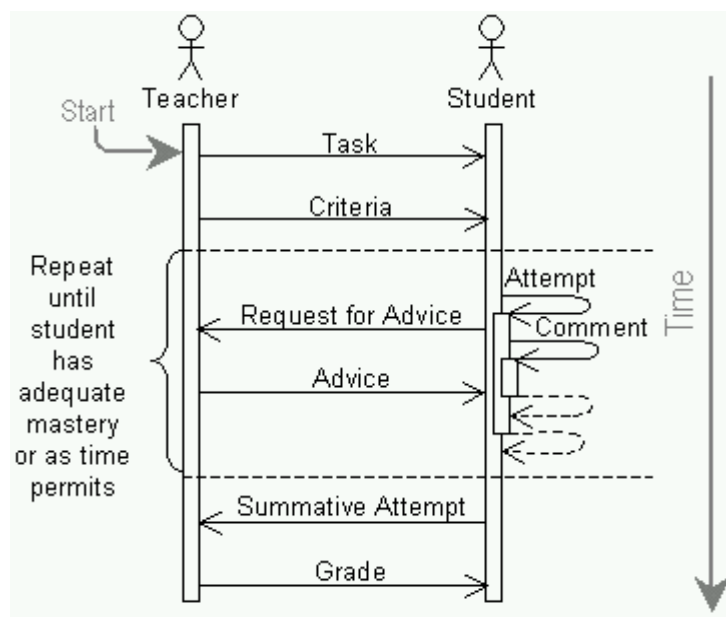


Figure 2: The advisory teacher/judicious student feedback loop

I used the advisory teacher/judicious student feedback loop also when students first learned about software development, and the review and quality assurance process. Once the allocated time for the judicious teacher/responsive student feedback loop had expired the students continued doing the software development tasks and reviewing their work. The students kept a portfolio of all their attempts and feedback. The initial attempts had feedback from the teacher; students reviewed their own subsequent attempts. The portfolio was evidence of their learning and final level of mastery for each software development task.

From my experience with the advisory teacher/judicious student feedback loop, some of the strengths of the loop are:

- The student is participating in making judgments and is more likely to acquire this skill;

- The teacher may be seen by the students as more approachable and less threatening;
- The students can work at their own pace after an initial encounter with the material.

Some of the weaknesses of the advisory teacher/judicious student loop I found to be:

- The students may initially flounder at making judgments and develop a negative opinion about studying the unit;
- The students don't have the subject expert knowledge to make accurate judgments;
- The teachers may be perceived to be shirking their 'traditional' responsibilities of teaching students how to do better and/or the subject content;
- The coverage of subject material may be significantly reduced, as the students are likely to need more time to attempt and evaluate a task.

The previous loops only required the students to evaluate themselves. The next loop engaged the students with their peers.

The judicious peer/responsive student feedback loop

The third feedback loop, judicious peer/responsive student (Figure 3), has the student's peer (or peers) commenting on and grading the student's work. The teacher's role may be to initiate the loop by setting a task or problem, and providing the criteria to the student and peer and/or input to the loop as determined by the students. Since many teachers are uncomfortable with students determining their own grade or the grade of others, this assessment activity can end with a summative assessment and grading by the teacher.

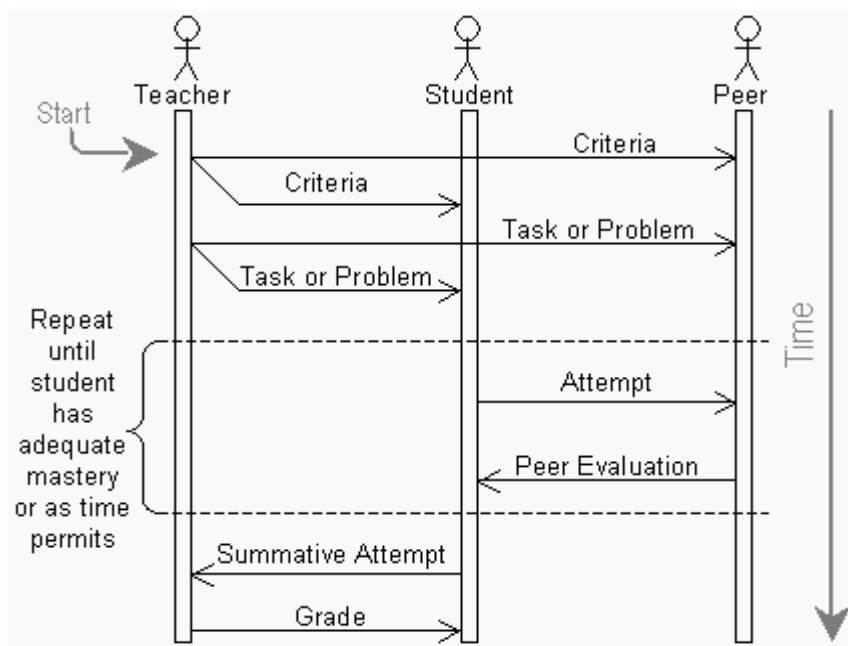


Figure 3: The judicious peer/responsive student feedback loop

I used the judicious peer/responsive student feedback loop when students were doing their second subject about software development. The students had experience with the previous loops and with many of the software development tasks. The novelty of the subject was the electronic support of the group approach to task completion and review and the use of project meetings rather than lectures.

final grade, were far removed from their comfort zone and put considerable energy into complaining rather than participating. Even though the individual tasks were designed to directly contribute to a group assignment, the number of students participating in the formative assessment fell to about two participants out of 32 as the session progressed.

From my experience with the self-determining student feedback loop, some of the strengths are:

- Successful students are more successful software developers, participating in the feedback process and performing well in formative and summative assessments;
- The teacher's role became explicitly one of a facilitator to learning, removing them from the judicious role.

Some of the weaknesses of the self-determining student feedback loop I found to be:

- Unsuccessful students become hostile;
- Teachers may feel they are too far removed from a comfortable teacher/student relationship.

Conclusion

Based on Scriven's original definition of formative evaluation, as it aligned nicely with the software development process, I formulated four feedback processes: judicious teacher/responsive student; advisory teacher/judicious student; judicious peer/responsive student; and self-determining student. In the judicious teacher/responsive student loop, the judicious teacher provides feedback so students can improve their work, and experience and respond to the review and quality assurance process. In an advisory teacher/judicious student loop, the judicious student can practice their review and quality assurance skills on themselves with the advice of the teacher. In a judicious peer/responsive student loop, the judicious peer, having been a judicious student, can perform more effectively as a reviewer of artefacts. In a self-determining student loop, the student becomes an independent learner; when presented with a problem the student is able to identify the appropriate path to take and assure the quality of their work along the way.

Experience to date with the feedback loops suggests that there are a number of things needed for their successful implementation. First, the students need to be informed explicitly of the feedback process to be used. Education, training, and communication are needed to enable the students to participate in the execution of the feedback loop. Communicating to the students which feedback loop is to be used and why is not enough. The students also need to be educated and trained in how best to participate in the feedback process.

Formative assessment requires good judgment to be made of the students' attempts at the assessment task. The judgment is communicated during a feedback process. To encourage students to learn quality assurance skills for successful software development we need to transfer the skills and knowledge not only of the subject content, but also of the feedback process. By the purposeful implementation of the feedback loops it is possible for students to become knowledgeable and skilled developers, comfortable with the review and quality assurance processes.

One of the strengths of the ideas presented lies in the link they have with the development of graduate attributes. One of the motivating and contributing factors to the development of the feedback loops and consequent subject design was the contribution to a student's development of generic graduate attribute skills. Concentration, at least within my current

school, of curriculum design on the discipline of computing without effort to explicitly include learning outcomes that are graduate-attribute focused makes the place for the feedback loops in such “discipline heavy” curriculum significant.

References

- Biggs, J. (1999). *Teaching for quality learning at university*. Buckingham, England: Society for Research into Higher Education and Open University.
- Black, P. (2000). Research and the development of educational assessment. *Oxford Review of Education*, 26(3/4), 407-419.
- Black, P. & William, D. (1998). Assessment in classroom learning. *Assessment in Education*, 5(1), 7-71.
- Bligh, D. A. (2000). *What's the use of lectures?* San Francisco, CA, USA: Jossey-Bass Publishers.
- Bloom, B. S., Hastings, J. T. & Madaus, G. F. (1971). *Handbook on Formative and Summative Evaluation of Student Learning*. New York: McGraw-Hill.
- Box, I. (2002). *Applying educational research to improve teaching and learning in information systems*. Informing Science and IT Education Conference, Cork, Ireland.
- Box, I. & Ferguson, J. (2002). *Object oriented software development: Step by step*. Sydney: Pearson Education.
- Boyd, B. L. (2001). Formative classroom assessment: learner focused. *The Agricultural Education Magazine*, 73(5), 1-8.
- Chase, C. I. (1999). *Contemporary assessment for educators*. New York: Addison-Wesley Publishers, Inc.
- Cowie, B. & Bell, B. (1999). A model of formative assessment in science education. *Assessment in Education*, 6(1), 101-116.
- Crooks, T. (1994). *Assessing student performance* (Vol. 8). Campbelltown, Australia: Higher Education Research and Development Society of Australasia Inc.
- Delbridge, A. (Ed.) (1997). *The Macquarie dictionary*. Sydney, NSW, Australia: Macquarie Library Pty Ltd.
- Entwistle, N. (1987). *Understanding classroom learning*. London: Hodder & Stoughton.
- Kanjee, A. (2000). Investigating formative assessment: Teaching, learning and assessment in the classroom. *Assessment in Education*, 7(1), 160-163.
- Pelligrino, J. W., Chudowsky, N. & Glaser, R. (Eds.). (2001). *Knowing what students know: the science and design of educational assessment*. Washington, DC: National Academy Press, free online version at <http://books.nap.edu/books/0309072727/html/index.html>.
- Rolfe, I. & McPherson, J. (1995). Formative assessment: How am I doing? *Lancet (North American Edition)*, 345, 837-839.
- Scriven, M. (1973). The methodology of evaluation. In B. R. Worthen & J. R. Sanders (Eds.), *Educational Evaluation: Theory and Practice* (pp. 60-104). Belmont, CA: Wadsworth Publishing Company.
- Scriven, M. (1991). Beyond formative and summative evaluation. In M. W. McLaughlin & D. C. Phillips (Eds.), *Evaluation and education: At quarter century* (Vol. 90, pp. 18-64). Chicago, Ill: National Society for the Study of Education : Distributed by the University of Chicago Press.
- Torrance, H. P. J. (2000). *Investigating formative assessment: Teaching, learning and assessment in the classroom* (Vol. 7). Buckingham ; Philadelphia: Open University Press.

Copyright statement

Copyright © 2003 Ilona Box: The author assigns to HERDSA and educational non-profit institutions a nonexclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The author also grants a nonexclusive licence to HERDSA to publish this document in full on the World Wide Web (prime sites and mirrors) on CDROM and in printed form within the HERDSA 2003 conference proceedings. Any other usage is prohibited without the express permission of the authors.