Title	Issues in software selection
Author(s)	Cheah Yin Mee & Cheung Wing Sum
Source	*REACT*, *1998*(2), 30-36
Published by	National Institute of Education (Singapore)

ISSUES IN SOFTWARE SELECTION

Review by Cheah Yin Mee and Cheung Wing Sum

INTRODUCTION

The launch of the IT2000 master plan in education in April 1997 has led to an increase in use of computers and related educational software for teaching and learning in Singapore schools. Singaporean teachers will have to take on yet another new role, that of software evaluators, before they can successfully use the new materials available. Although teachers have traditionally been evaluating print material and textbooks as part of their teaching duties, the business of software evaluation presents new and challenging issues. This article will discuss a number of issues that are related to software evaluation and review some of the implications for teachers in Singapore schools.

SOFTWARE EVALUATION

A major problem in software evaluation lies in the software. By and large, English language CD-ROM software produced for language learning is often developed in English-speaking countries for monolingual speakers of the language. Software packages for English as a second language (ESL) speakers are not easily available; bilingual software packages with English and the appropriate native languages are even fewer. In Singapore, because local software production is still in its infancy, there is no choice but to rely on foreign countries to supply the software. An obvious problem that can arise is the possibility of cultural incompatibility where content and pedagogy are concerned. Another problem is the variety of English and the level of language used.

Gill, Dick, Reiser, & Zahner (1992) reported that the most prevalent approach to selecting software is to review evaluations published by software evaluation services. Independent reviewers do most of these evaluations through the use of a checklist. These reviewers are required to make subjective decisions about the accuracy of the content and the effectiveness of the package for the classroom. In addition, they also have to rate its technical strengths and weaknesses. Gill et al. has questioned the reliability of such reviews, but this approach is still used because of its convenience. It is also the fastest way to provide some information about any new piece of software.

This same strategy for software selection is popular in Singapore too, but the problem is magnified many times because many of the reviews and evaluations are done in contexts that are outside Singapore, notably in Western, native-speaking English countries. For example, many software programs developed in the United States tend to use examples that are familiar to American audiences. These may pose problems to learners in Singapore and in other societies who are not as familiar with the examples provided. Reviewers who are unfamiliar

with these contexts or who are not writing with these readers in mind will not point out these problematic issues in their reviews.

Software reviews that are available to guide teachers in their selection are almost always not directed at our teachers nor at our school population. Although there may be generic learning situations and activities that are applicable across situations, these are limited in their use, and reviews (except Singaporean ones when available) do not point out the exceptions to Singapore teachers. Thus software rated as excellent elsewhere may not be useful in other contexts because the teaching-learning situations may not support its use.

Sam Cheah

Similarly, the pedagogy or learning objectives inherent in such software may not be appropriate in these contexts. Needless to say, the students who end up using the software are also different from the audience the writers had in mind when they first conceived of the package.

Because of this, there is a need for teachers themselves or for local reviewers to review the software with the local context in mind. But when this happens, reviewers tend to rely on established checklists of criteria since none have been developed locally. The relevance of these criteria is questionable, but more of this later.

SOFTWARE EVALUATION VS. SOFTWARE REVIEWS

A number of issues can be identified in relation to the very notion of software evaluation. The first has to do with what the process of evaluation entails. For some, evaluation is a process of description and appraisal of software by teachers or reviewers while for others the process includes pre- and post-testing and observations of student use (Centre for Educational Research and Innovation, 1989). The former notion of evaluation (a process of description and appraisal of software alone) is probably the most popular. However, it is the latter process, which includes pre- and post-testing, that provides the most useful information for teachers. These two strategies also reflect a formative vs. summative approach to evaluation. In a formative evaluation according to Knussen, Tanner, & Kibby (1991 p.14) the concern is with progress towards achieving the goals of an

educational innovation and thus the process of description and appraisal is deemed adequate. A summative evaluation is concerned with the effectiveness of a programme upon completion in relation to its stated aims, and this makes pre- and post-testing a core component of the process. Most publications feature formative evaluations of software - summative evaluations are rarely available to the public.

SOFTWARE EVALUATORS

The issue of who the evaluators are is also important. Heller (1991) pointed out that evaluation is often carried out by a wide range of individuals and groups ranging from students, classroom educators, administrators and librarians as well as members of professional associations. These reviews are, as expected, subjective views that are much influenced by the individual's knowledge, expertise and interest. A study by Jolicoeor & Berger (1986) showed that there was little correlation between recommendations given by two review services, while Callison and Haycock (1988) found weak correlation between program ratings given by teachers and pupils. Heller (1991, quoting Burt, 1985) and Knussen, Tanner, & Kibby (1991, quoting Lawton, 1980) identified teachers as the most appropriate persons to evaluate the software because they are the end users, and they know best the context and condition under which the software will be used. However, teachers rarely have the time to do extensive reviews of software before using it in the classrooms. They, therefore, tend to rely on published reviews as guidelines.

CRITERIA FOR EVALUATION

What to evaluate in a software program is yet another issue. Reviews could focus on the technology, the content, or the pedagogic presentation or all three. Naturally, the reviewers' strengths and interests play an important part in determining the focus of the review, but this will also mean that the review will only be of use to certain groups. This brings us back to the issue of the criteria to be used in software evaluation. The criteria used for evaluation will reflect the predominant concerns of the reviewer or the context for which the review is to be used and this often results in an emphasis on different criteria. An obvious solution to this is to have a comprehensive list of criteria covering all major aspects, but this is not practical as the task will become more time consuming for evaluators and users. For example, the New South Wales Department of Education in Australia prepared a set of criteria for teachers, which had seventeen evaluation areas, which in turn generated one hundred and twelve criteria! (Rowe, 1993).

Technology

While the criteria for technical aspects may be applicable across many contexts, we need to remember that the technology available in different contexts also varies. However, the same cannot be said for content and for pedagogical presentation. Where content is concerned, we need to take the subject or the academic discipline into consideration when evaluating the software. For example, software dealing with history or geography requires a different set of evaluation criteria than

those dealing with say science or language. Content software (for subjects like science or history) should contain up-to-date and correct information, and this software could be used for reference alone. However, information is less important in language software, but opportunities to use language meaningfully are vital. Needless to say, the accuracy of the language is also important. Language software also needs to provide good integration of the four skills in their activities, a criterion which merits lower priority in a content software. So far, most checklists ignore such differences in disciplines. Instead, a generic list of criteria is produced and is used to rate the effectiveness of all software programs. This is not helpful to teachers looking for ways of evaluating discipline-specific software. The very task of allotting a number of points to each criterion also makes a farce of the whole evaluation process. The sum of the points does not necessarily tell you how effective a software is going to be in the classroom.

Content and Instructional Design

Next, the content and instructional design of each software program have to be taken into consideration since what is produced in one cultural context may not always fit in with the educational curriculum or the cultural practices of teaching and learning in another context. For instance, the centralised curriculum in Singapore means that all schools must follow the established syllabus and use prescribed materials. There is often little scope for the inclusion of activities and materials that are not in the syllabus. Software programs produced elsewhere do not always have matching curricula content. Thus, the usefulness of a software is in the end dependent upon the creativity of the teacher-user, rather than what the reviews report.

Pedagogic Presentation

In addition, activities that are part of the software program may require pedagogical approaches that are not practised here or are not popular with teachers given the constraint of the curriculum and the class size. Or the software may favour certain pedagogic presentation styles that are culturally inappropriate here. A good example is found in *Reading Galaxy* or in *Reading Blaster* where a hip game host takes the children though the activities. Children unfamiliar with TV game hosts will have a problem relating to the software. This issue of cultural fit is important when considering any material for use in classrooms. Software evaluation procedures neglect this aspect and instead treat the activity as a neutral and culture-free process.

WHAT CRITERIA TO USE

Our position has always been not to prescribe criteria to teachers, although it is true that teachers need guidelines to help them select appropriate software. In our investigation into this issue of the type of software criteria that Singapore teachers use, we found that these are the top four criteria among pre-service teachers (Cheah & Cheung, in preparation).

Software Criteria	Pre-service teachers' ranking
User-friendly	1
Interesting	2
Appropriateness	3
Easy to understand	4

Table 1: Pre-service teachers' ranking of software criteria

These findings tell us that our pre-service teachers may prefer these criteria but their interpretation of each of these criteria also vary. For instance, the term *"user-friendly"* which is, by now, an overused and imprecise term, drew a number of interpretations ranging from *"simple to use"*, *"simple instructions"*, *"help easily available"*, *"easy to navigate"*, *"useful icons"* to *"no technical knowledge required"*. This in turn illustrates another problem of relying on criteria listing as a guide.

Many researchers have developed lists of such criteria for evaluating software (see Gros and Spector, 1994; Heinich, Molenda, Russell & Smaldino, 1996). The only criteria listing generated by a research team is Bitter and Wighton's work (1987), and while these lists do provide useful insight into the process of evaluating software, they are often too lengthy and tend to be less than helpful for busy teachers. Bitter and Wighton's work for instance produced 22 separate criteria.

However, teachers may find it useful to heed the following four categories of minimum criteria proposed by Roblyer, Edwards, and Havriluk (1997 p 120):

1. Required instructional design and pedagogy: Does it teach?
2. Required for content: Is it correct?
3. Required for user flexibility: Is it "user-friendly"?
4. Required technical soundness: Does it work correctly?

CONCLUSION

With these issues in mind, it is clear that selecting a piece of software based on commercial or other published criteria can be problematic because of the different contexts and populations that the software has to be put to use in. What teachers value and need in each context can be very different, and this is dependent upon the set curriculum, the social and cultural contexts for teaching and learning, the established pedagogical procedures, and also the degree of freedom teachers have to innovate.

IMPLICATIONS

Bearing these issues in mind, there are several implications for teachers to consider when selecting software.

1. ***When reading any write-up about software, it is always useful to ask if the write-up is a review, i.e. a description only or a proper evaluation.***
   Many reviews can be biased, or not very useful if they only describe the content of the software.

2. ***Keep in mind where you found the write-up.***
   Is it from a journal which looks at educational issues? Think about where the reviewer may be coming from, and try to read the article with their goals in mind.

3. ***Keep your teaching and learning context in mind when reading reviews/ evaluations.***
   An award-winning software package may not work for you if you do not agree with its teaching and learning philosophies or if its content does not fit your curriculum goals.

4. ***Try and list your objectives for using a piece of software.***
   Then try and match these to those in the software. Your criteria should matter most.

5. ***Remember there is no such thing as an ideal piece of software.***
   Awarding a number of points to various criteria as a strategy to select a piece of software can be meaningless because the best software need not get the highest points if your criteria are not properly set out in the first place.

6. ***Always keep the cultural issue in mind.***
   This could be a major stumbling block to learning.

7. ***Ensure that the basic computer hardware requirements for the software are available when trying out the software.***

8. ***Develop your own set of criteria for selecting software.***

SOURCES

Bitter, G. & Wighton, D. (1987). The most important criteria used by educational software evaluation consortium. *The Computing Teacher*, **14** (6), 7-9.

Burt, C. (1985). Software in the classroom: A form for teacher use. *The Computing Teacher*, **12**.

Callison, D., & Haycock, G. (1988). A methodology for student evaluation of educational microcomputer software. *Educational Technology*, **28** (1), 25-32.

Centre for Educational Research and Innovation. (1989). *Information Technologies in education. The quest for quality software*. Paris: OECD.

Cheah, Y. M., & Cheung, W.S. (in preparation) *Software evaluation criteria-Perspectives from Singapore pre-service teachers.*

Gill, B. J., Dick, W., Reiser, R. A., & Zahner, J. A. (1992). A new model for evaluating instructional software. *Educational Technology*, 39 - 44.

Gros, B., & Spector, J. (1994). Evaluating automated instructional design systems: A complex problem. *Educational Technology*, **34** (5), 37-46.

Heinich, R., Molenda, M., Russell, J., & Smaldino, S. (1996). *Instructional Media and Technologies for Learning*. Prentice Hall: Englewoods Cliffs, NJ.

Heller, R. S. (1991). Evaluating software: A review of the options. *Computers and Education*, **17** (4), 281-291.

Jolicoeur, K., & Berger, D. E. (1986). Do we really know what makes educational software effective? A call for empirical research on effectiveness. *Educational Technology*, **26**, (12), 7-11.

Knussen, C., Tanner, G. R., & Kibby, M. R. (1991). An approach to the evaluation of hypermedia. *Computers Education*, **17**, (1), 13 -24.

Lawton, D. (1980). The politics of curriculum evaluation. In D. Tawney (ed.) *Curriculum Evaluation Today; Trends and Strategies*. London: Schools Council Research Studies, Macmillan.

Roblyer, M., Edwards, J., & Havriluk, M. (1997). *Integrating Educational Technology into Teaching*. Upper Saddle River, NJ: Prentice Hall, Inc.

Rowe, H. (1993). *Learning with Personal Computers: Issues, Observations, and Perspectives*. Australian Council for Educational Research: Hawthorn, Victoria, Australia.