EVOLUTION OF EDUCATIONAL SOFTWARE EVALUATION: INSTRUCTIONAL SOFTWARE ASSESSMENT

Servet Bayram
Computer Education & Instructional Technologies
Marmara University , Istanbul, Turkey
E-mail: sbayram@marmara.edu.tr
http://aef.marmara.edu.tr/bilgisayar/english/servet.htm

Albert P. Nous
Instruction and Learning, School of Education
University of Pittsburgh, Pgh. PA. USA
E-mail: apnous2@pitt.edu
http://www.education.pitt.edu/people/AlbertNous/

Abstract

Many popular terms such as software description, software review, software evaluation and lastly software usability used by design, development and evaluation experts in the field. However, such terms used interchangeably by researchers and developers are syntactically and semantically different due to their conceptual backgrounds and the technical procedures taken in the development or evaluation process. A simple search of the basic terms, software selection, software description, software review, software evaluation, software usability and software assessment in this day and age of electronic databases would yield a large number of "hits" or articles. There are different definitions for such terms. The definitions are used loosely and interchangeably in activities by the researchers and web page developers. The term *software assessment is used* to encompass the set of terms described in this paper. This allows for evaluating the contributions of specific characteristics to the effectiveness of instructional software anatomy.

Keywords: Instructional software, assessment, evaluation, review, and usability

Evolution of Educational Software Evaluation: Instructional Software Assessment

The Nature of Educational Software Evaluation

Over the past twenty years there has been a large increase in the number of instructional software's titles available for classroom use. A substantial industry exists for developing software for a particular curriculum area, grade level, and machine. This industry relies on the production of software evaluation guidelines and recommendations. Guides and descriptions of commercially available programs are generated profusely and distributed widely for the education market (Bianchi, 1994; Wilson, 2000). One might think that even present demands for evaluation would be a causal influence on the improvement of software quality. The problem of this study is to analyze the literature and contribute to a better understanding of research associated with the evolution, current use, and implications for software evaluation in education.

There is no doubt that the infusion of web-based learning products occupies a significantly large market share of the medium used today. Multimedia or hypermedia computer applications also hold promises for instruction and assessment of learning (Kumar et al., 1995). But these products may be nothing more than "new packages for old ways of learning" and the system of evaluating the new media has not changed since the floppy disk came onto the market (Surry, 1995, 1998; Wilson, 2000; Scantlebury et al., 2001). Media and their attributes have important influences on the cost or speed of learning but only the use of adequate instructional software methods will influence learning. This generalization requires that we reexamine the goals for software evaluation.

The most often stated primary goal of evaluating instructional software is to determine if it can be used effectively in a particular classroom (Reiser & Kegelmann, 1994). Traditionally, the teacher first determines if software objectives are consistent with, and complementary to, the objectives already set for that classroom. The objectives identified by commercial software authors may not equivocate to what a specific program actually teaches or assesses (Bonekamp, 1994). Oftentimes, unintended program objectives, discovered after some experience in using the software, replace existing course-of-study objectives. With a growing abundance of educational software, teachers face difficult selection decisions. Often, their choices are made under the constraints of limited funds for computer materials and tight classroom schedules. In these circumstances, teachers seek educational software they feel most likely to produce positive learning outcomes (Hawley et al., 1997).

When computer use in education took hold in the early eighty's, the Educational Products Information Exchange (http://www.epie.org) reported that only five percent of available educational software could be rated as "exemplary." Along with non-EPIE software evaluations for the period 1980-1984, EPIE suggested that the overall level of quality had progressed no further than the lower end of EPIE's "recommended with reservations" rating range (Staff, 1985, 1994). In another study it was reported that "...not all supposedly good, educational software is used as intended. And even when it is, the predefined objectives often are far from being achieved. Not enough is known about how learning takes place (Caftori, 1994, p. 62)." Little has changed in this regard over the years. If technology is to become a vital part of teaching process, it must take its place with all of the other aids the teacher has to meet the needs of each student (Basden, 2001). While the timeliness of the conclusions might be dated, its appropriateness still applies today.

Software evaluation, as a problem-solving activity in itself, could be viewed as one in which the nature of the problem is constantly changing. In this view, the products of evaluations are "stills" from a moving picture, rather than defined portraits of cognition. Hence, opportunity exists for international cooperation in the review, evaluation, access to, and research in the use of educational software. Software quality depends on a number of interrelated, evolving variables and should therefore be considered in the light of past and future developments (CERI, 1989, pp. 105-09). The last point emphasizes our concern about the evolution of software evaluation as highlighted in this paper. It also points out the nature of the "industry" and needed research associated with software evaluation, usability and implementation.

In general, current software evaluation approaches range from large numbers of evaluators in organized networks to the individual teacher using her or his own discretion; but all tend to be normative in nature (Surry, 1998; Forgan & Weber, 2001; Bangert-Drowns, 2002). That is to say, individuals rate or evaluate software according to their strength of agreement on statements about the software, or they give written opinions on various aspects of a program. A problem for consumers of such evaluation is that current procedures tend to elicit predominantly subjective judgments. The weaknesses identified often with evaluation practices include their normative nature, subjectivity, lack of reliability, and difficulty in obtaining an overall impression (Haugland & Shade, 1990; Reiser & Kegelmann, 1994; Wilson, 2001; Scantlebury et al., 2001). To judge the validity of reviews or evaluations, publishers were urged to include responses to questions such as: Who were the reviewers? What experiences have they had with the subject matter, intended student population, and instructional techniques of the software? How many students, and under what conditions, used the software? What were student backgrounds and abilities? How did reviewers collect the information used for their conclusions and judgments? How were quality ratings determined? And, what evidence showed that the ratings of these reviewers were valid? But validity, user numbers, data collection, rating determination, etc., are concerns that emanate primarily from a focus on the adult organizational membership and not student cognition and actions, or behavioral content (Scriven, 1990; Basden, 2001). That is to say evaluation is in service of the organization, and its administrative managers, rather than the individual learner. As understanding of classroom use of computers and of learning process increases, techniques for developing educational software are still evolving (Squires & McDougall, 1994; Robb & Susser, 1999; Forgan & Weber, 2001).

Almost all of the software evaluation and usability checklists contain a considerable number of items concerning the hardware needed to run the program, the presence and quality of documentation and printed support materials, the topic area and content of the program, its ease of use and reliability of operation, and the use of color, graphics and sound in the program. Relatively few evaluation studies have been conducted on the use of computers in education and of the learning outcomes of the different modes of educational software. That seems to be the domain of theses and journal publications. There is consequently the need for many more such studies and research into what constitutes quality software (Presby, 2001). Such research should be engendered by the software industry in partnerships with the academics.

From the above perspective, the nature and the meaning of software evaluation, usability and theirs' specific settings was discussed. Many popular terms such as software description, software review, software evaluation and lastly software usability used by design, development and evaluation experts in the field. However such terms used interchangeably by researchers and developers are syntactically and semantically different due to their conceptual backgrounds and the technical procedures taken in the development or evaluation process.

The Words that Confuse

The four words description, review, evaluation and usability combined with the words software, technical, pedagogical, or educational, generate a set of 16 possible terms hindering a simple across-the-board comparison of software. In the literature, for example, these terms are used interchangeably: software description, technical description, pedagogical description, software review, technical reviews, educational reviews, software evaluation and software usability or technical usability.

The first term, *software description*, is used often to represent an objective and informative description of software "package." Software description has as its commercial the display of objective information by publishers for individuals and administrators involved with the justification of expenditures for acquisition of software for instruction. Software description is often further delineated into two subsets: *technical* and *pedagogica*l. A *technical description* usually includes bibliographic information such as author, publisher, and distributor, and delineates parameters such as memory requirements, cost, the hardware required to run the package, and peripherals such as mouse, light pen, CD-ROM drive, graphics pad, and joy-stick (CERI, 1989; Bianchi, 1994; Greiner, 2002). *Pedagogical description* typically represents information such as subject area, type of software (i.e., tutorial, drill and practice, simulation, discovery, etc.), target audience, educational objectives, a description of the user support materials, and a statement of the flow and sequence of the program. Pedagogical software descriptions are presented from Greiner in 2002.

The second term, *software review,* is usually reserved for a more critical appraisal by one or more teachers and experts. Reviewer opinions may, or may not, be backed by descriptions of how the software was used by the reviewer or by observations of usage of the software in the classroom (Buckleitner, 2002). Software review elicits personal viewpoints, recommendations, and thoughts about how a particular package would be used in the classroom. Individuals who rate software are concerned with administrative or management issues primarily for feasibility of use based on prior experiences with students as end users. Reviewers often use a checklist or review sheet and follow a set procedure in reviewing the software (Heller, 1991; Reeks, 2002).

The third term *software evaluation* is used for a rendered judgment about the value and usefulness of a piece of software by means of quantitative and qualitative measurements and methodologies. Software evaluation also elicits judgments about value, usefulness and appropriateness of software for student use. Determination of success or failure to meet student needs is based on limited use with students before a judgment to buy is rendered. (Wilson, 2000; Bengert-Drowns, 2002). Quality of instruction and congruence with curriculum objectives is paramount in statements about software use for classroom learning and possible research on cognition. Evaluation techniques might include pre- and post-testing or detailed observation of student use. Software evaluation can be either formative or summative (Troutner, 2002).

Both software reviews and reports of software evaluations normally include, what are referred to as, *technical* and *educational* reviews. The *technical review* is often conducted using a checklist of features rendering a judgment about aspects of the software, such as its robustness and reliability under normal conditions, the design of its display, its ease of use, procedural instructions and prompts, user control and, if applicable, its quality of color, sound, graphics and animation (CERI, 1989; Buckleitner, 2002). *Educational review* is a judgment about the value of a piece of software as a teaching or learning tool. For example, the content is accurate and educationally valuable, the level of difficulty is appropriate to the specific audience, and the software achieves its stated objectives (Littauer, 1994; Lindroth, 2002). Examples of educational review are found in Troutner (2002).

One senses a continuum of activity between simple technical description and educational review. There are different definitions for description, review and evaluation, and although use of the software is different across the globe, in the United States, these terms are often used synonymously as a judgment about the value, worth, appropriateness and usefulness of a piece of educational software. Also, they are used loosely and interchangeably in activities ranging from sales descriptions of software in magazines to in-depth critical appraisals conducted by academic researchers. In some form of written expression the technical and pedagogical comments are generally provided. Although the same standards for technical excellence appear to apply across cultural, international and curriculum boundaries, software still needs to be assessed its importance in the curriculum. Thus a piece of educational software might be excellent in itself as far as its technical or even its educational aspects are concerned but be of no importance in the curriculum. Oftentimes many reviewers are sold the "package" but it remains on the "shelf." This need for curriculum fit is particularly important when software is being assessed for use in countries other than the country of origin of software (Boekamp, 1994).

Software review and *evaluation* can be thought of as a continuum with various shades of meaning and interpretation depending on the methodologies used and consequently with various levels of objectivity and usefulness. These may be expressed in (1) a factual description of the software without any evaluation comments; (2) evaluations which are the outcomes of studies by academic researchers adhering to rigorous norms of scientific inquiry; (3) reviews which reflect the personal viewpoints of the reviewers; or (4) reviews which are the consensus of a group of examiners. There are fewer reviews and evaluations, which include data collected during the observation of students in the classroom (e.g. Heller, 1991) than reviews, which have been written without such observation (e.g. Haugland & Shade, 1990). There are reviews written by classroom teachers and "experts" (e.g. Surry, 1995, 1998). There are even fewer reviews and evaluations which include the comments of students (Squire & McDougall, 1994; Wilson, 2000; Liaupsin, 2002).

The last term *software usability* can be defined as a measure of the ease with which a product can be learned or used, its safety, effectiveness and efficiency, and attitude of its users towards it. (Preece, et al, 1994). Most usability testing with end user products is formative in nature and can be divided into four types : exploratory, assessment, validation and comparison. (Rubin,1994). These types correspond with formative/summative goals of usability testing. Web usability, technical usability, pedagogical usability and educational usability are among popular terms in the usability testing and usability engineering area. Software usability as an assessment type may serve a number of different purposes: to improve existing product; to compare two or more products; to measure a system against a standard or a set of guidelines (Sugar, 2001).

Since software value to users varies, we sought a term that would recognize and maintain this diversity yet allow us to "tax" these evaluations and generate a "common currency for communication." Hereafter, for sake of brevity, the authors will use the term *software assessment* to encompass the set of terms described above.

Promises in Instructional Software Assessment

A simple search of the basic terms, software selection, software description, software review, software evaluation, software usability and software assessment in this day and age of electronic databases would yield a large number of "hits" or articles. For instance, on November 17, 2003, we used Alta Vista search on the Internet to see how many related links there are for such terms especially for children. Two mounts later same search was done again by the way of Yahoo and Google search engines. The following Table 1 reports such search keywords and their results.

Table 1. Educational Software Links for Children on the Net

Key words/ Search Engine	**Alta Vista** Nov.17, 2003	**Yahoo** Jan. 15, 2004	**Google** Jan.10, 2004
Educational software 'selection for children'	144,807	512,000	331,000
Educational software 'description for children'	140,902	451,000	355,000
Educational software 'review for children'	246, 126	1,040,000	642,000
Educational software 'evaluation for children'	158,100	439,000	327,000
Educational software 'usability for children'	14,076	47,100	19,100
Educational software 'assessment for children'	173,100	484,000	346,000
Total	877,111	2,973,100	2,020,100

Based on search engines result for "educational software '….' for children" there are millions web links on the net as seen on Table 1. As it mentioned before there are different definitions for the terms software description, review, evaluation and usability on the web pages. Such definitions are also used loosely and interchangeably in activities by researchers and web page developers. It is not until one examines each type for its objectives, primary recipient, purposes, foci, and measurement concerns that we begin to see subtle distinctions. The authors used the term *software assessment* to encompass the set of terms described in this paper. Table 2 focuses on the software assessment types and their specific features.

Table 2. Types of Software Assessment

Type	**Objective**	**Used by**	**Purpose**	**Focus on**	**Measurement**
Description	Objective/ Descriptive information	Publishers Individuals Administrators	Commercial Display	Bibliographic machine information	None
Review	Views Class users	Teachers	Acquisition Curriculum fit	Pedagogical and technical aspects	Potential
Evaluation	Value, Efficiency Appropriateness	Experts Researchers	Student achievement	Pedagogical utility Technical aspects	Formative

	Further development Effectiveness	Teachers			Summative
Usability	Utilization, Efficiency Functionality Effectiveness Visibility, Safety	Experts Researchers Individuals	User friendliness Satisfaction, Success Achievement Productivity	Bibliographic and machine information, Pedagogical/Technical aspects; Instructional view, GUI Interactions	Formative Summative

Software description, as an assessment type, is observed often at the "front end" of the educational process. It is used while software is still on the vendor's "shelf" or selected by an administrator or "corporate educational buyer" before commitments and applicability are determined for a particular subject, grade, class, or student. Administrator or curriculum coordinator activity predominates in the decision to make software available for review and evaluation.

Software review, as an assessment type, occurs at the "front door" of the classroom when software is taken off the vendor's shelf and previewed by the teacher. It is this teacher who then decides whether or not to recommend or actually purchase the software on the basis of what he or she values. Teacher activity dominates this process.

Software evaluation, as an assessment type, occurs "within" the confines of the learning environment; students are given access to software and use is observed. Judgment about software "worth" and further use is determined relative to student achievement or intellectual gains. Extensions to research initiatives are often made and include some measure of the action of students as they progress through learning activities mediated by the computer. As such, an evaluation of this type goes beyond considering sensory modalities, audio and visual effects, input requirements, level of difficulty or degree of confusion about goals and objectives.

Software usability, as an assessment type combines most of the above. Also, it is the measure of the quality of the users experience when interacting with something. It is the analysis of the design of a product or system in order to evaluate the match between users and a product or system within a particular context. Usability evaluation should be considered to be a dynamic process throughout the life cycle of the development of a product. Conducting evaluation both with and without end-users significantly improves the chances of ensuring a high degree of usability of product for the end user. Usability assessment has five considerations: (1) ease of learning, (2) efficiency of use, (3) process ability, (4) error frequency and security, and (5) subjective satisfaction. It is also allows developers to obtain and appreciate a user perspective of their product.

An overview of the components, functions and limitations of the human cognitive system provides a framework for understanding why some educational software that "looks good" fails to produce positive outcomes. Unfortunately, nearly all software evaluation systems are heavily weighted on computer-related dimensions such as error-handling, aesthetic considerations, such as the quality of screen displays, sound, touch, and content related issues of scope, sequence, and accuracy. Although important, these characteristics do not address what we know about how students learn (Reeks, 2002). There was no study focusing on learners and their specific interaction with the "*anatomy of the software*." From this point of view, the critical question: "Anatomy of Instructional Software Assessment: Mirror Image of Usability Testing or Evolution of Software Evaluation?" needs to be answered for further studies.

Conclusion

Evaluation studies must be conducted on the use of computers across the curriculum, of the learning outcomes using the different modes of educational software, how these modes serve specific instructional needs, and certainly, the degree and substance of human-computer interactions. Specific characteristics of educational software must be related to their combined effectiveness in promoting learning. And the validity of reviews as measures of the effectiveness of educational software must be established (Gong et al., 1992; Bangert-Drowns, 2002). While some indicators of quality are lacking in the majority of software available, it is important that teachers and schools be aware of, begin to look for, and expect them in the software they are asked to purchase. Similarly it is important that not only software authors but also curriculum developers be aware of such indicators. Instructional software should be designed in the context of what is known about how student learn. And the software should be fully tested to be sure that it "teaches" effectively and efficiently in specific learning activities. The likelihood of obtaining high-quality software this way is considerable greater than through the procedure of having a computer programmer go off to write instructional software (Caftorio, 1994). But again, adhering only to guidelines in the development of software does not necessarily ensure successful infusion of the package into the curriculum.

Continued research is needed to explore further the aspects of instructional software, which are attractive to students and apply them in software development. There is a need for further examination of screen display design, the graphic-user interface, and information access by humans to identify the criteria for assessing the impact of instructional software in different subject areas and for evaluating the contributions of specific characteristics to the effectiveness of instructional software. This same charge can be made for World-Wide-Web access. Computer software is seen as a "boon" for mass education but, as Hendry and King (1994) point out, students maintain their uniquely constructed understandings despite being taught scientific explanation. Teachers should not expect software to complement their transmission views of teaching (Forgan & Weber, 2001). If particular software does not meet our standards of probable learning effectiveness, it is unlikely to prove satisfactory in everyday teaching situations.

Finally, the above question presented before the conclusion now deserves an answer. The Anatomy of Instructional Software Assessment: Mirror Image of Usability Testing or Evolution of Software Evaluation? An admittedly premature answer is that it is old wine (software evaluation) in a new bottle (delivery techniques: usability testing). However, forming partnerships among the instructional designers, researchers, learning and instructional psychologist, cognitive and computer scientists, educators, trainers and of course learner can lead a new, substantive change in the quality of the wine (assessment), and therefore a new vintage (anatomy of software).

References

Bangert-Drowns; R. L. (2002). Teacher ratings of student engagement with educational software: An exploratory study; *Educational Research & Development Technology, 50*(2), 23.

Basden, J. C. (2001). Authentic tasks as the basis for multimedia design curriculum, *T.H.E. Journal, 29*(4).

Bianchi, A. (1994). The irresistible customer questionnaire, *Inc, Vol. 16*(12), 97-103

Bonekamp, L. W. F. (1994). Courseware evaluation activities in Europe. *Journal of Educational Computing Research, 11*(1), 73-90.

Buckleitner, W. (2002). JumpStart advanced kindergarten, *Scholastic Parent & Child, 10* (2), 53.

Caftori, N. (1994). Educational effectiveness of computer software. *T.H.E. Journal, 22* (1), 62-65.

Center for Educational Research and Innovation (CERI) (1989). *Information Technologies in Education: The Quest for Quality Software*. Organization for Economic Co-Operation & Development (OECD), Paris.

Forgan, J. W. & Weber, R. K. (2001). Software for learning, Part two: Choosing what is best for your child. *The Exceptional Parent;* 31(7), 60-65

Gong, B., Venezky, R., & Mioduser, D. (1992). Instructional assessments: Lever for systemic change in science education classrooms. *Journal of Science Education & Technology, 1*(3), 157-176.

Greiner, L. (2002). OneOnOne Computer Training; *Computer Dealer News, 18(*11); 25.

Haugland, S. W. & Shade, D. D. (1990). *Developmental evaluations of software for young children*. Delmar Publishers Inc. Albany, New York.

Hawley, C. , Lloyd, P.., Milkulecky, L.., & Duffy, T. (1997, March). Workplace simulations in the classrooms: The teacher's role in supporting learning. Paper presented at the annual meeting of the AERA, Chicago, IL.

Heller, R. (1991). Evaluating software: A Review of the options . *Computers & Education,17* (4), 285-91

Hendry, G. D. & King, R. C. (1994). On theory of learning and knowledge: Educational implication of advances in neuroscience. *Science Education 78* (3), 239..

Kumar, D. D., White, A. L. & Helgeson, S. L. (1995). A study of the effect of Hypercard and traditional pen/paper performance assessment methods on expert/novice chemistry problem solving. *Journal of Science Education and Technology, 4* (3), 157-176.

Liaupsin, C. J. (2002). The comprehensive evaluation of a self-instructional program on functional behavior assessment. *Journal of Special Education Technology, 17*(3), 5.

Lindroth, L. (2002). Blue ribbon technology, *Teaching Pre K-8, 33*(3), 22.

Littauer, J. (1994). A "How to.." on using courseware in the classroom. *T.H.E. Journal, 22*(1), 53-54.

Preece, J. ; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S., & Carey, T. (1994). *Human-Computer Interaction*, Workingham, England: Addison-Wesley.

Presby, L. (2001). Increasing Productivity in course Delivery, *T.H.E. Journal, 28*(7).

Reeks, A. (2002). Tools for learning, *Parenting, 16*(2), 134.

Reiser, R. A. & Kegelmann, H. W. (1994). Evaluating instructional software : A review and critique of current methods, *Educational Technology Research & Development, 42*(3), 63-69

Robb, T. N. and Susser, B. (1999). The life and death of software: Examining the selection process. The 4th Annual JALT CALL SIG Conference on Computers and Language Learning May, 21-24, 1999. Kyoto Sangyo University; Kyoto, Japan.

Rubin, J. (1994). *Handbook of usability testing*: How to plan, design, and conduct effective tests. Wiley., NY.

Scantlebury, K. C. ; Boone, W. J. ; and Butler, K. J. (2001). Design, validation, and use of an evaluation instrument for monitoring systemic reform. *Journal of Research in Science Teaching. 38*(6), 646-662.

Scriven, M. (1990). The evaluation of hardware and software, *Studies in Educational Evaluation*, *16*, 3-40.

Squires, D. & McDougall, A. (1994). *Choosing and Using Educational Software: A Teachers' Guide*. Falmer Press, Taylor & Francis, Inc., 1900 Frost Road, Suite 101, Bristol, PA 19007. (ERIC Document No: ED 377 855).

Staff. (1985, April/May/June). What's in the educational software pool? *MICROgram*, 1-4.

Staff. (1994). Defining software for the curriculum. *Syllabus*, 8(1), 16-17.

Sugar, W. A. (2001). What is a good about user-centered design? Documenting the effect of usability sessions on novice software designers. *Journal of Research on Computing in Education, 3(*3), 235-250.

Surry, D. (1995, Nov/Dec). Software-Plus Review. The AECT*, Techtrends*, 36-38.

Surry, D. (1998, March). Biolab Frog & Biolab Pig, Math for the Real World, Peterson’s Interactive Career & College Quest,. Facts on File World News CD-ROM and Educator’s Essential Internal Training System. The AECT, *Techtrends ,*8-11.

Troutner; J. (2002). Bookmark it: Best new software, *Teacher Librarian; 30*(1), 53.

Wilson, H. W. (2000). How to evaluate educational software, *Curriculum Review;* 39(1), 15.

Wilson, H. W. (2001). Beyond data mania, *Leadership, 31*(2), 8-36.