

# Technology *in* Action

VOL. 3 • ISSUE 1 • NOV 2007

## The Software $\sqrt{\quad}$ -List

### Evaluating Educational Software for Use by Students with Disabilities

**Randall Boone & Kyle Higgins**  
University of Nevada Las Vegas

There are few validated tools with which parents and teachers can confidently evaluate software for students with disabilities. This *Technology in Action* presents one promising tool for this purpose—the Software  $\sqrt{\quad}$ -List.

*Two students are viewing the same instructional video placed within a multimedia software program designed to teach or reinforce the meaning of the verb “to lose.” A mime walks to the center of the screen and smiles. Then her expression changes. She begins to search through her pockets and pat her shirt and pants as though looking for something. She looks behind herself and to each side of herself to no avail. And then, in a final act of futility, she rubs a tear from one eye. Her expression is a very sad face.*

*The video pauses before going on to the next scene and the students respond separately and independently to a simple question: “What was happening to the person in the video?” The typical learner responded that the mime clearly had lost something and was trying to find it again. The student with a learning disability, however, became fixated on the final act of the mime (i.e., rubbing a tear from her eye) and indicated that the mime was sad.*

*Two students with similar needs for vocabulary development responded very differently to a typical multimedia educational intervention. From an instructional perspective one student got it right and the other got it wrong.*

The lesson from this vignette is simple. The psychology of user design and instructional focus of educational software is much more complex than many educators imagine, especially when it comes to students with disabilities (Boone & Higgins, 2007). Was the student who associated the concept of “sad” with the video wrong? Or was the software content ambiguous and unclear?

This conundrum became evident in the situation described above because the software was being used in a formative evaluation setting. The software developers were working directly with students who were typical of the target audience,

asking questions and taking notes as the students interacted with the program. Had this particular phase of development been skipped or had the participants not been typical of the target audience, this problem with the instructional video likely would have gone unnoticed and uncorrected, causing problems for the non-typical learners who later would encounter it at school (Higgins & Boone, 2006).

This *Technology in Action* looks at how the *Software √-List* addresses this issue. Read on to learn about the development of this tool and how you can use it to evaluate software for students with disabilities.

## Evaluating Educational Software for Students with Disabilities

Because commercial educational software often forms the nucleus of a technology-based instruction program in special education classrooms, obtaining quality educational software that is targeted for specific audiences (e.g., learning disabilities, English language learners, intellectual disabilities) is vital. Educators rely on commercial software developers to produce quality educational software for classroom use, with the assumption that the software has been designed to meet the unique learning needs of the students for whom it is targeted (Williams, Boone, & Kingsley, 2004).

Unfortunately, the results of research focusing on the instructional design of commercial educational software are not encouraging. In fact, much educational software is developed without consideration of key educational factors that may affect learning for students with disabilities (Ager, 1986; Boone & Higgins, 2007; Boone & Higgins, 2005; Geisert & Futrell, 1995; Golden, 2002; Higgins, Boone, & Williams, 2000; Larsen, 1995; Lockard, Abrams, & Many, 1997; Neuman, 1991; Williams, Boone, & Kingsley, 2004; Zane & Frazer, 1992).

This dearth of forthright evaluative information from the educational software community often leaves teachers and parents in the dark when it comes to selecting and using educational software with their children. Often, parents and educators find that the software they have purchased is not adaptable, does not teach what was expected, or does not support the learning processes that are occurring (Higgins, Boone, & Williams, 2000; Williams, Boone, & Kingsley, 2004).

Because there are so few resources, parents and educators often perform their own evaluations. When doing so, they should take the following into consideration:

- Intended use of the software and whether or not the software can help achieve those objectives.
- Content of the software and

whether or not the software supplements teacher-led instruction.

- Instructional presentation of the software and whether or not the software meets the principles of universal design (i.e., multiple representations of content, multiple means of expression and control, and multiple forms of engagement).
- Ease of use by teacher and student.
- Documentation and support provided.
- Technical adequacy of the software.

This is not an easy task and there are few validated tools with which parents and teachers can evaluate software for students with disabilities confidently. One such tool—the *Software √-List*—provides a solution.

## Development of the Software √-List

The *Software √-List* was developed with the assistance of a panel of specialists in the areas of special education and technology, special education teachers, and parents of students with disabilities. The following procedures were used:

- Review of the literature.
- Development of a software evaluation blueprint.
- Formative item review.
- Content validation.
- Second review.
- One-on-one field testing.
- Development of final product.

A description of each follows. [Note: Construct validity of all items in the *Software √-List* has not yet been completed.]

### Review of the Literature

Phase One of the project focused on the collection and review of the literature to identify characteristics of appropriate software for students with disabilities. Specifically, the research literature concerning technology for students with learning disabilities, intellectual disabilities, emotional disabilities, and physical disabilities was searched, along with the literature on early childhood disabilities. All age groups were included in the literature search (i.e., pre-K through high school).

This comprehensive literature review produced a list of research-based characteristics of appropriate software for specific disability groups and age groups. These characteristics were broken into dimensions and items. For example, the dimension of Student Needs Based Upon Disability Area included such items as:

- Communicates relevant features of a task.
- Manual dexterity not an important skill.

The review of the literature revealed more than 200 items with the following dimensions: accessibility, accuracy, authenticity, branching, directions, documentation, evalua-

●

**Often, parents and educators find that the software they have purchased is not adaptable, does not teach what was expected, or does not support the learning processes that are occurring.**

●

tion, feedback, independence, individualization, instructional levels, instructional strategies, interactivity, motivation, modalities, practice, prompts, record keeping, scope and sequence, screen design, self-correction, sound, student options, teacher options, technical issues, and tutorials.

### Evaluation Blueprint

The blueprint for the *Software √-List* was created with consideration for the learning characteristics of students in each of the major disability areas (i.e., learning disabilities, intellectual disabilities, emotional disabilities, and physical disabilities).

Early childhood also was included as a category at this level. The blueprint focused on these categories, as well as age groups (i.e., pre-K, elementary, middle school, and high school), and general software characteristics for all students with disabilities.

All items on the *Software √-List* were stated as positive statements and included a five-point Likert scale for parents and teachers to respond to the statements. All items were written and designed according to the general guidelines recommended for developing items for a Likert format (Crocker & Algina, 1986; Worthen, Borg, & White, 1993).

### Formative Item Review

A panel of experts reviewed the blueprint for the *Software √-List*, paying particular attention to the items in the Dimensions column. At this stage of the formative evaluation process, the responses from the experts were used to cluster items on the evaluation tool, correct errors, and add or delete items. The items were mapped onto specific dimensions as a result of the input from the experts, and the first formative revision of the software evaluation tool was conducted.

A senior research associate to the project and an expert in the area of instrument validation reviewed the resulting blueprint for accuracy

and completeness, paying particular attention to blueprint items. The items were checked for clarity, grammar, spelling, and level of readability (Crocker & Algina, 1986). Additional items and dimensions also were provided.

### Content Validation

The expert panel then reviewed the revised *Software √-List* blueprint using a Content Validation Worksheet. The panel was asked to assess the relevance of every item in each dimension with respect to the use of educational software with students with disabilities. Specifically, the panel was asked to place each item on the evaluation tool into one of the dimensions listed on the worksheet. They also were asked to identify any items they believed were not relevant or did not belong in a specific dimension, and they were encouraged to provide additional items and dimensions.

Expert panel responses were coded and entered into an item-dimension placement database. The database provided a basis for determining panel member's agreement or disagreement with the original blueprint. To be included in a dimension, at least three out of the five panelists had to have placed an item in a particular dimension.

The software evaluation tool was revised again taking into consider-

ation the expert panel recommendations concerning content validation. This was the second iteration of revision in the formative evaluation process.

### Second Review

Formative feedback was solicited from four groups of teachers who worked with students with disabilities:

- Special educators who provide services to students with disabilities full time in self-contained environments (e.g., a resource room or a self-contained classroom).
- Special educators who provide services to students with disabilities through a collaborative consultation model in general education.
- General educators who provide services to students with disabilities through a collaborative consultation model in general education.
- Educational computing strategists.

Formative feedback also was solicited from two groups of parents:

- Parents whose children receive special education services full time in a self-contained environment.
- Parents whose children receive special education services in the general education environment with consultation services provided by a special educator.

The responses of the parents and teachers were used to cluster items, correct errors, and add or delete dimensions or items. The responses were compared to the responses

from the expert panel and revisions were made. This was the third iteration of the formative evaluation process in developing the *Software √-List*.

### One-on-One Field Testing

The software evaluation tool blueprint was field tested in a one-on-one fashion with teachers and parents. The teachers and parents used the software evaluation tool with a piece of educational software targeted for use with students with disabilities. The purpose of this phase of the evaluation was to identify and correct any major problems with the tool when it was applied to software. During this phase the goal was to determine if the user:

- Understood the instructions.
- Knew what to do.
- Could interpret what was expected.
- Could read and understand all of the textual material.

The parents and teachers were encouraged to talk out loud as they applied the *Software √-List* to the software.

In this one-on-one activity, project personnel asked each individual teacher or parent about any problems he or she experienced when applying the software evaluation tool to software. Notes were taken concerning the kinds of problems encountered, and questions asked

by the teachers and parents were recorded.

All data collected in the one-on-one formative evaluation were coded into data source categories (Smith & Ragan, 1999). All revisions at this stage of the evaluation were tied directly to problems experienced by the teachers and parents in the actual usage of the *Software √-List*. The one-on-one interaction with the parents and teachers constituted the fourth and final formative evaluation iteration for the project.

### Development of Final Product

The *Software √-List* consists of six separate forms, including the General Form for all students with disabilities and five additional forms that correspond to the following categories:

- Learning disabilities.
- Intellectual disabilities.
- Physical disabilities.
- Emotional disabilities.
- Early childhood.

[*Note:* These forms are available at the end of this *Technology in Action*.]

Seven categories of concern were identified, but they do not necessarily occur on each form. The categories are:

- Instruction.
- Directions and documentation.

The *Software √-List* consists of six separate forms, including the General Form for all students with disabilities and five additional forms that correspond to the following categories:

- Learning disabilities.
- Intellectual disabilities.
- Physical disabilities.
- Emotional disabilities.
- Early childhood.

- Feedback and evaluation.
- Content.
- Individualization options.
- Interface and screen design.
- Accessibility.

The suggested protocol for using these forms includes first using the General Form and then selecting the appropriate second form based on the specific disability of the student or students with whom the software might be used. This is because some items that are deemed *good* or *appropriate* for a student with one disability might not be effective or appropriate for another student with a different disability. We have found that an item on the General Form (i.e., Graphics should not be overwhelming or distracting), while also appearing on the Learning

Disabilities Form, is inconsistent or in conflict with items on the Emotional Disabilities Form, which states that auditory and visual effects should be emphasized (Boone & Higgins, 2007). Similarly, pacing for students with emotional disabilities is suggested to be “fast-paced,” while time constraints are discouraged for students with learning disabilities, and time constraints for students with intellectual disabilities are suggested to be regulated by the teacher.

### Using the Software √-List

Each of the six forms in the *Software √-List* is constructed similarly and contains the following elements:

- Disability heading.
- Categorized subheadings for areas of concern appropriate to the specific disability area.
- Specific items under each categorized subhead that are stated as positive attributes of educational software.
- Decision boxes for user determination if specific items are found in the software being evaluated.

The text box, *Four Parts of the Software √-List Forms*, shows these elements.

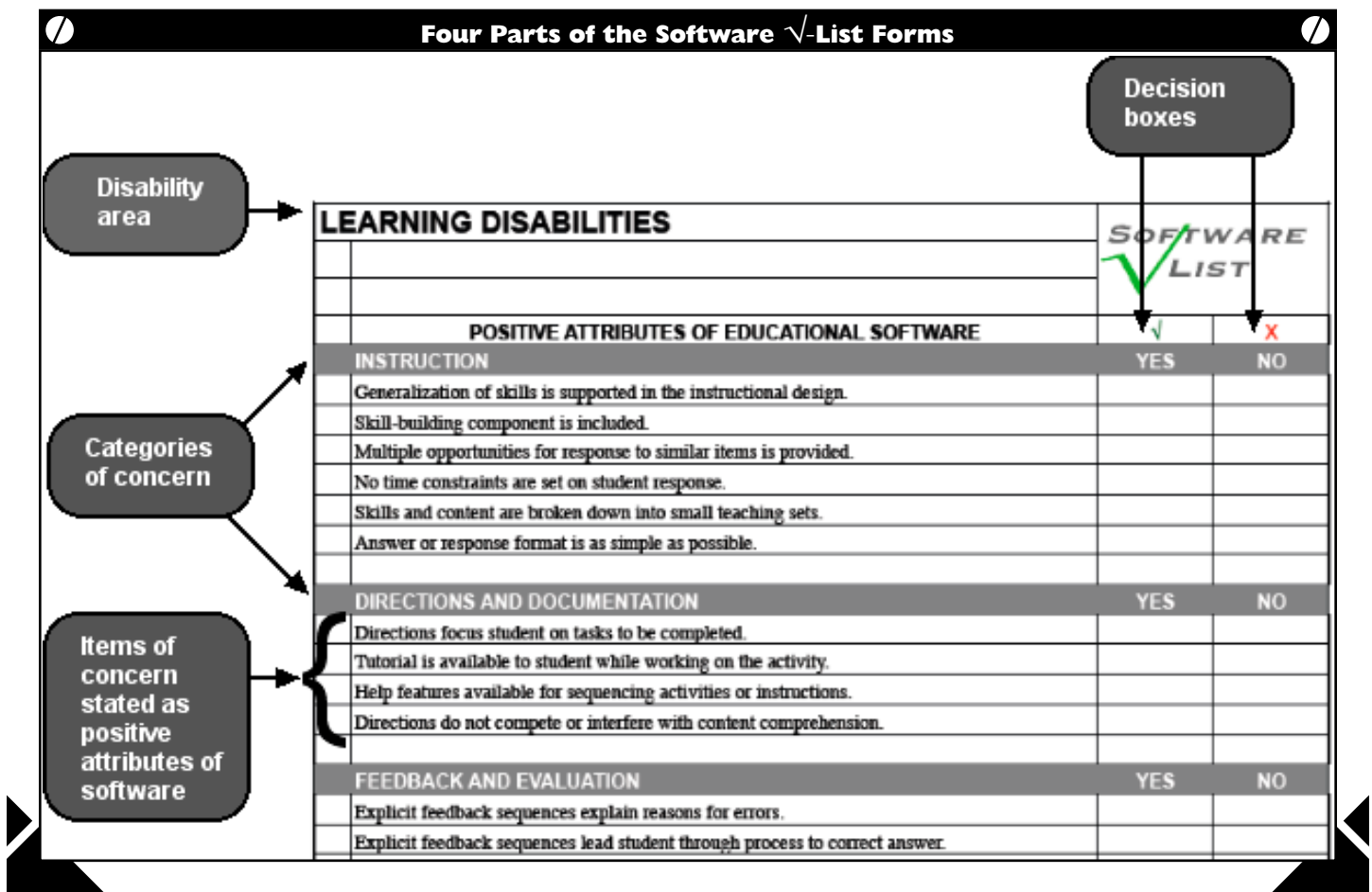
### Rubric

Currently there is no formalized or codified process or set of rules for decision making based on the information garnered from the *Software*

✓-List. This is because software decisions need to be made individually for each child. Also, it is quite common that a piece of software will not present the evaluator with the opportunity to respond “Yes” or “No” to all items of concern listed in a particular category. Our suggestion for using the forms follows a simple protocol:

- Go through the software once by yourself without using the form.
- Observe your student or child using the software if possible, again without using the form.
- Fill out the General Form while using the software by yourself or with a colleague. Search through different areas of the software to determine if the software contains (Yes) or does not contain (No) the attributes stated in the items of concern in each category. Some items will not be applicable to each piece of software or you will not be able to make the determination. Simply leave the decision box for those unanswered items empty. For an example, see the text box, *Unfilled Decision Boxes*.
- Select the appropriate disability form (e.g., Learning Disabilities Form) and complete it.

You will make an informed but subjective decision as to whether a piece of educational software is good enough to employ it. Not only will you have to weigh the number of “Yes” vs. “No” determinations that were made, but you also must pay attention to the content of those items and the category of concern under which they were listed. This is where the educational goal for the student and the student’s own strengths and weaknesses enter as new variables in the decision-making process.



For example, under the Instruction category of the General Form, there are four specific strategies listed:

- Outlining.
- Questioning.
- Repetition.
- Using advance organizers.

If a student has had success with one of these in the past, then the absence or presence of that particular strategy could be a prime determiner in deciding whether or not to use the software. The items on the form provide an organized outline to support the evaluation process.

This is not to say that comparing the number of “Yes” and “No” decisions for items of concern may provide an unrealistic or invalid first impression in the evaluation process. It simply cannot stand alone as the only determiner for using or excluding a piece of software for a particular student.

**E-Learning Principles**

Current research-based instructional design principles for e-learning (e.g., multimedia and online instruction) include the following (Clark & Mayer, 2003):

- **Multimedia.** Using words and graphics can be more effective than using just words alone.
- **Contiguity.** Printed words should be placed near the corresponding graphics.
- **Coherence.** Adding extraneous sounds, pictures, or words can hurt learning, even if the additional material is interesting.
- **Modality.** Information presented as spoken words, rather than on-screen text, is more educationally effective. This suggests that graphics with audio might be more effective than graphics with associated text.
- **Redundancy.** Presenting words simultaneously in both text and verbatim audio narration can hurt learning.

While these principles are well documented and based on sound cogni-

tive learning theory, the research guiding them reflects outcomes of typical learners. Therefore, because we may find conflict between these widely accepted principles of design and the learning needs of many students with disabilities, these principles cannot be factored into the evaluation rubric without a caveat or two.

There is potential for discrepancies between the positive attributes itemized on the *Software √-List* and some of these design principles. For example, item four under the Instruction category on the General Form reads, “ideas and concepts are shown in multiple representations (e.g., text and graphics).” This item also is recommended on the Learning Disabilities Form. It is concurrent with the multimedia principle mentioned above and it makes sense when viewed from a universal design for learning perspective (Rose & Meyer, 2002). For an example, see the text box, *Positive Attribute*

Unfilled Decision Boxes		
POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE	✓	✗
INSTRUCTION	YES	NO
Learning cues or prompts support complex tasks.	✓	
Lessons can be interrupted and returned to without starting over.		✗
Problem solving, reflection, and creativity are promoted over rote learning.		
Ideas and concepts are shown in multiple representations (e.g., text & graphic)	✓	
Small teaching sets simplify content.		✗
Strategies for transfer of skills are provided.		

*Item in Conflict with a Multimedia Instructional Design Principle.*

However, this design practice has the potential for being used inappropriately if it doesn't follow the contiguity principle (words are placed near corresponding graphics) and the modality principle (graphics with audio are more effective than graphics with text) (Clark & Mayer, 2003). The multiple representation of text-to-speech technology that is widely used with students with disabilities (Anderson & Anderson, 2005; Lewis, 2005; Montali & Lewandowski, 1996; Okolo, 2005; Wepner, 1990)

●

**Even with the best of evaluation tools, making decisions about educational materials, especially computer software and online instruction, remains a difficult task.**

●

also seems to be at odds with the redundancy principle (e.g., don't use simultaneous verbatim text and speech).

Again, the assumption that must be drawn from this discussion leads back to a software evaluation that is rule-based but not rule-bound. That is, the *Software √-List* provides a matrix of rules for analysis of the software being considered for a particular student, but it does not require a specified outcome. That path might be easy, but it simply would not be accurate.

Positive Attribute Item in Conflict with a Multimedia Instructional Design Principle	
<b>POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE</b>	
<b>INSTRUCTION</b>	✓ YES
Learning cues or prompts support complex tasks.	
Problem solving, reflection, and creativity are promoted over rote learning.	
Ideas and concepts are shown in multiple representations (e.g., text & graphic)	
Small teaching sets simplify content	
A self-correction process leads student to the answer	

**Seemingly at loggerheads with the Redundancy instructional design principle which suggests that use of simultaneous verbatim text and speech output can hurt learning.**

## A Final Thought on Software Evaluation

Even with the best of evaluation tools, making decisions about educational materials—especially computer software and online instruction—remains a difficult task. Adding to the difficulty is the fact that many teachers and parents hold low expectations for educational software.

Cooper (1999) uses the example of a dancing bear to explain this situation, saying: “the wonder isn’t that the bear dances well, but that the bear dances at all” (p.26). For those teachers and parents who use educational software, the wonder isn’t that the text-to-speech system speaks the text clearly and in a consistent, logical order, but that the computer can speak the text at all. It is easy to miss the fact that the spell-check function doesn’t really help the student become a better speller, it just identifies the words that are incorrectly spelled. In other words, educators often are willing to adapt their teaching to fit inflexible software and students are willing to continue working with software even after their work has been lost due to an ambiguous file-saving process (Boone & Higgins, 2005). These things are accepted as part of the price for seeing the bear dance.

## Using the Software √-List

The six forms that comprise the *Software √-List* follow. They may be reproduced and used in unlimited quantity as long as the copyright attribution is retained on all copies (Copyright © 2004. Randall Boone and Kyle Higgins).

## References

- Ager, A. (1986). Performance contoured programming: A structure for micro-computer-based teaching of individuals with severe learning difficulties. *Journal of the Association for Programmed Learning*, 23(2), 130-135.
- Anderson, K., & Anderson, C. (2005). Integrating technology in standards-based instruction. In D. L. Edyburn, K. Higgins, & R. Boone (Eds.). *Handbook of special education technology research and practice*, pp. 521-544. Whitefish Bay, WI: Knowledge by Design Publications.
- Boone, R., & Higgins, K. (2005). Designing digital materials for students with disabilities. In D. L. Edyburn, K. Higgins, & R. Boone (Eds.). *Handbook of special education technology research and practice*, pp. 481-492. Whitefish Bay, WI: Knowledge by Design Publications.
- Boone, R., & Higgins, K. (2007). The role of instructional design in assistive technology research and development. *Reading Research Quarterly*, 42(1), 135-140.
- Clark, R. C., & Mayer, R. E. (2003). *E-Learning and the science of instruction*. San Francisco, CA: Pfeiffer.
- Cooper, A. (1999). *The inmates are running the asylum*. Indianapolis, IN: Sams.
- Crocker, L., & Algina, J. (1986). *Introduction to classical and modern test theory*. New York: Harcourt, Brace, Jovanovich.
- Geisert, P. G., & Futrell, M. K. (1995). *Teachers, computers, and curriculum*. Boston: Allyn and Bacon.
- Golden, D. C. (2002). Instructional software accessibility: A status report. *Journal of Special Education Technology*, 17(1), 57-60.
- Higgins, K. & Boone, R. (2006, January). *Evaluating educational software for use by students with disabilities in special and general education*. Paper presented at the 2006 Hawaii International Conference on Education, Honolulu.
- Higgins, K., Boone, R., & Williams, D. (2000). Evaluating educational software for special education. *Intervention in School and Clinic*, 36(2), 109-115.
- Larsen, S. (1995). What is “quality” in the use of technology for children with learning disabilities? *Learning Disabilities Quarterly*, 18(2), 118-130.
- Lewis, R. (2005). Classroom technology for students with learning disabilities. In D. L. Edyburn, K. Higgins, & R. Boone (Eds.). *Handbook of special education technology research and practice*, pp. 325-334. Whitefish Bay, WI: Knowledge by Design Publications.

Software -List

# GENERAL FORM FOR ALL STUDENTS WITH DISABILITIES

<b>POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE</b>	✓	✗
<b>INSTRUCTION</b>	<b>YES</b>	<b>NO</b>
Learning cues or prompts support complex tasks.		
Lessons can be interrupted and returned to without starting over.		
Problem solving, reflection, and creativity are promoted over rote learning.		
Ideas and concepts are shown in multiple representations (e.g., text & graphic)		
Small teaching sets simplify content.		
Strategies for transfer of skills are provided.		
A self-correction process leads student to the answer.		
Independent exploration is available and is encouraged.		
Outlining is used as a instructional strategy.		
Questioning is used as an instructional strategy.		
Repetition is used as a instructional strategy.		
Advance organizers, summaries, or outlines used as instructional strategies.		
Instructional transitions from one level to another include content overlaps.		
Important points or concepts are visually highlighted (e.g., bold).		
Software adapts to student input and branches to appropriate instructional level.		
<b>DIRECTIONS AND DOCUMENTATION</b>	<b>YES</b>	<b>NO</b>
Directions are provided on screen.		
Directions are simple.		
Instructions can be repeated for student.		
Relevant features of a task are clearly communicated.		
Teacher documentation is easy to understand.		
Specific instructional objectives are outlined for the teacher.		
Prerequisite skills for all aspects of the software are listed.		
State and national standards are addressed in documentation.		
Hardware requirements are clear.		
<b>FEEDBACK AND EVALUATION</b>	<b>YES</b>	<b>NO</b>
Feedback is consistent.		
Feedback is immediate.		
Feedback is obvious.		
Feedback is positive and relevant to action taken by student.		
Feedback on incorrect responses indicates where or how error occurred.		
Student responses are recorded automatically by the software.		
Final performance evaluation is provided.		
Evaluation provides information on learner strengths and weaknesses.		
Learner progress records can be viewed or printed.		

## Software ✓-List

# GENERAL FORM

## FOR ALL STUDENTS WITH DISABILITIES (page 2)

POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE	✓	X
CONTENT	YES	NO
Content is free of grammatical errors.		
Content information is accurate.		
Content is free from bias (e.g., ethnic, gender)		
Content is relevant to the curriculum.		
Realistic, real-life situations are used in activities and content.		
Difficulty level of content is appropriate		
Content is interesting.		
Audio is appropriate to instructional content.		
A variety of difficulty levels are available.		
INDIVIDUALIZATION OPTIONS	YES	NO
Student can control rate, amount, and sequence of presentation.		
No time constraints are set on student response.		
Content and activities can be modified by the teacher.		
Rate, amount, and sequence of content can be controlled by the teacher.		
INTERFACE AND SCREEN DESIGN	YES	NO
Screen design features are consistent and clear.		
Screen is uncluttered.		
Navigation elements are clearly available on all screens.		
Graphics are not overwhelming or distracting.		
ACCESSIBILITY	YES	NO
Alternate input devices work with the program.		
Digital speech output is provided for all text.		
Verbal directions have corresponding on-screen text available.		
Minimal keyboarding skills are required.		
Text spacing assures readability.		
Accuracy of input is minimized (e.g., typing errors can be corrected)		
Navigation though the program is simple.		

Copyright © 2004. Randall Boone and Kyle Higgins

Software ✓-List		
<b>EARLY CHILDHOOD</b>		
<b>POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE</b>	✓	X
<b>INSTRUCTION</b>	YES	NO
Exploration and open-ended discovery is encouraged.		
Cause-and-effect processes are highlighted.		
Trial-and-error discovery is promoted.		
Multiple correct answers are possible for activities.		
Activities encourage creative play.		
Problem solving activities are included.		
Orientation is on the process rather than on a product.		
Social interaction activities are included.		
<b>DIRECTIONS AND DOCUMENTATION</b>	YES	NO
Instructions are simple and are available as graphics or sound.		
Directions are either iconic or aural (i.e., no written directions)		
<b>FEEDBACK AND EVALUATION</b>	YES	NO
Child can complete tasks independently with little instruction or teacher monitoring.		
<b>CONTENT</b>	YES	NO
Content and activities are developmentally appropriate.		
Content is presented in appropriate entry level, then increasing levels of challenge.		
No violence is explicit or implicit in the content or activities.		
Content accurately portrays aspects of a child's world.		
<b>INDIVIDUALIZATION OPTIONS</b>	YES	NO
Program branches to appropriate new content or tasks.		
<b>INTERFACE AND SCREEN DESIGN</b>	YES	NO
Menus and navigation elements are pictures and icons.		
Choices are available and easy to select.		
<b>ACCESSIBILITY</b>	YES	NO
Hand-eye coordination skills are appropriate.		
Keyboard requirements are minimal.		

Software <input checked="" type="checkbox"/> -List		
<b>LEARNING DISABILITIES</b>		
POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE	✓	✗
<b>INSTRUCTION</b>	<b>YES</b>	<b>NO</b>
Generalization of skills is supported in the instructional design.		
Skill-building component is included.		
Multiple opportunities for response to similar items is provided.		
No time constraints are set on student response.		
Skills and content are broken down into small teaching sets.		
Answer or response format is as simple as possible.		
<b>DIRECTIONS AND DOCUMENTATION</b>	<b>YES</b>	<b>NO</b>
Directions focus student on tasks to be completed.		
Tutorial is available to student while working on the activity.		
Help features available for sequencing activities or instructions.		
Directions do not compete or interfere with content comprehension.		
<b>FEEDBACK AND EVALUATION</b>	<b>YES</b>	<b>NO</b>
Explicit feedback sequences explain reasons for errors.		
Explicit feedback sequences lead student through process to correct answer.		
Procedures are consistent for correction and evaluation.		
<b>CONTENT</b>	<b>YES</b>	<b>NO</b>
Variety of difficulty levels are available.		
Ideas and concepts shown in multiple representations (e.g., text, video, graphics).		
<b>INDIVIDUALIZATION OPTIONS</b>	<b>YES</b>	<b>NO</b>
Diagnostic and prescriptive strategies provided for placement of student in program.		
Self-pacing provided for student.		
Motor skills and manipulative acts are kept simple.		
<b>ACCESSIBILITY</b>	<b>YES</b>	<b>NO</b>
Motor skills and manipulative acts are kept simple.		

Copyright © 2004. Randall Boone and Kyle Higgins

Software ✓-List		
<b>INTELLECTUAL DISABILITIES</b>		
POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE	✓	✗
<b>INSTRUCTION</b>	<b>YES</b>	<b>NO</b>
Task-analysis model used in instructional design.		
Sequential progression with controlled transitions used in instructional design.		
Easy discrimination of tasks is promoted by instructional design.		
Repetition of content or skills is utilized for all tasks.		
Unlimited practice is provided.		
Instructional tasks are of short duration.		
Relevant stimuli are clearly identified and emphasized (e.g., color codes, large size)		
Tutorial agent is available for feedback and information.		
Initial instruction focuses on acquisition of new skills.		
Initial tasks designed for high probability of success by students.		
Internal locus of control is promoted.		
<b>DIRECTIONS AND DOCUMENTATION</b>	<b>YES</b>	<b>NO</b>
Entry requirements clearly specified in documentation or help system for teacher.		
<b>FEEDBACK AND EVALUATION</b>	<b>YES</b>	<b>NO</b>
Personalization of feedback (e.g., student's name is used)		
<b>CONTENT</b>	<b>YES</b>	<b>NO</b>
Content and activities include real-life contexts.		
Content and activities are age appropriate.		
Content and activities include a variety of formats.		
Photo-realistic graphics and video are used to increase generalization.		
Abstract concepts are illustrated by realistic and concrete graphic depictions.		
Functional words and symbols are incorporated.		
<b>INDIVIDUALIZATION OPTIONS</b>	<b>YES</b>	<b>NO</b>
Reading level can be varied by the teacher.		
Time control for responses can be regulated by the teacher.		
<b>INTERFACE AND SCREEN DESIGN</b>	<b>YES</b>	<b>NO</b>
Visual display is free of unnecessary distractions.		
Navigation and menu system is consistent.		
Audio prompts are provided.		
Icons and pictures used instead of words for menu and navigation choices.		

Copyright © 2004. Randall Boone and Kyle Higgins

Software ✓-List		
<b>PHYSICAL DISABILITIES</b>		
POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE	✓	✗
INSTRUCTION	YES	NO
Diagnostic and prescriptive strategies are provided for placement of student in program.		
ACCESSIBILITY	YES	NO
Screen magnification programs work in conjunction with the software.		
A screen reader program can access all text displayed by program.		
Input speed does not affect accuracy of response.		
Output of alternate file formats for student reports provided (e.g.,text for Braille readers)		
Graphic resolution allows magnification without distortion.		
Sound used as for navigation and menu selections.		
Alternate input devices are accommodated by the software.		
Software function is not altered by use of alternate input devices (e.g., single switch).		

Copyright © 2004. Randall Boone and Kyle Higgins

Software ✓-List		
<b>EMOTIONAL DISABILITIES</b>		
POSITIVE ATTRIBUTES OF EDUCATIONAL SOFTWARE	✓	✗
INSTRUCTION	YES	NO
Activities are fast-paced, requiring student to respond quickly.		
Problem-solving models are incorporated into the instructional design.		
DIRECTIONS AND DOCUMENTATION	YES	NO
Explicit goals are provided to the student.		
FEEDBACK AND EVALUATION	YES	NO
Feedback is immediate, continuous, and positive.		
Score keeping is available.		
INDIVIDUALIZATION OPTIONS	YES	NO
Varied performance levels are available to students.		
INTERFACE AND SCREEN DESIGN	YES	NO
Auditory and visual effects are emphasized (e.g., video game look-and-feel).		

Copyright © 2004. Randall Boone and Kyle Higgins

- Lockard, J., Abrams, P. D., & Many, W. A. (1997). *Microcomputers for twenty-first century educators*. New York: Longman.
- Montali, J., & Lewandowski, L. (1996). Bimodal reading: Benefits of a talking computer for average and less skilled readers. *Journal of Learning Disabilities, 29*, 271-279.
- Neuman, D. (1991). Learning disabled students' interactions with commercial courseware: A naturalistic study. *Educational Technology, Research, and Development, 39*(1), 31-49.
- Okolo, C. (2005). Interactive technologies and social studies instruction for students with mild disabilities. In D. L. Edyburn, K. Higgins, & R. Boone (Eds.). *Handbook of special education technology research and practice*, pp. 623-641. Whitefish Bay, WI: Knowledge by Design Publications.
- Rose, D., & Meyer, A. (2002). *Teaching every student in the digital age: Universal design for learning*. Alexandria, VA: Association for Supervision and Curriculum Development.
- Smith, P. L., & Ragan, T. J. (1999). *Instructional design*. Upper Saddle River, NJ: Merrill.
- Wepner, S. (1990). Computers, reading software, and at-risk eighth graders. *Journal of Reading, 34*(4), 264-268.
- Williams, D., Boone, R., & Kingsley, K. (2004). Teacher beliefs about educational software: A Delphi study. *Journal of Research in Teacher Education, 36*(3), 213-229.
- Worthen, B. R., Borg, W. R., & White, K. R. (1993). *Measurement and evaluation in the schools*. New York: Longman.
- Zane, T., & Frazer, C. G. (1992). The extent to which software developers validate their claims. *Journal of Research on Computing in Education, 24*(3), 410-419.
- Research and development for this evaluation tool was supported in part by a grant from the U.S. Department of Education, Office of Special Education Programs (Steppingstones of Technology, Phase 1).*

**TAM Technology in Action** is a publication of the Technology and Media Division of the Council for Exceptional Children. **Cynthia Warger**, Editor. Editorial Board: **Christine Appert, Margaret Bausch, Gayl Bowser, John Castellani, Kyle Higgins, Brenda Heiman, Penny Reed and Windy Schweder**. TAM President: **Brenda Heiman**. Address correspondence to: C. Warger, P.O. Box 3836, Reston, VA 20195, [cwarger@msn.com](mailto:cwarger@msn.com). For information on becoming a TAM member, visit the TAM website at [www.tamcec.org](http://www.tamcec.org).

© 2007 Technology and Media Division (TAM)

TECHNOLOGY AND MEDIA DIVISION  
Council for Exceptional Children  
1110 No. Glebe Rd., Suite 300  
Arlington, VA 22201-5704



Technology and Media Division  
A Division of The Council for Exceptional Children  
1110 North Glebe Road, Suite 300  
Arlington, VA 22201-5704