



Differences in the educational software evaluation process for experts and novice students

Hatice Sancar Tokmak, Lutfi Incikabi and Tugba Yanpar Yelken
Mersin University

This comparative case study investigated the educational software evaluation processes of both experts and novices in conjunction with a software evaluation checklist. Twenty novice elementary education students, divided into groups of five, and three experts participated. Each novice group and the three experts evaluated educational software selected by the novice groups in accordance with the checklist. Data were collected through focus group interviews, classroom observations and document analysis. Evaluation processes were analysed through thematic comparisons. The results showed that the expert-novice agreement rate was as low as 48%, with novice students tending to grade the software higher. While the experts used a systematic approach, including understanding and assessing each criterion, supporting the process with literature and evaluating the software as a whole, novice students lacked such methods, indicating a need for additional training and development.

Introduction

Many expert-novice comparison studies have been conducted in the teaching field. For example, Chi, Feltovich and Glaser (1981) investigated novice and expert representations of physics problems related to knowledge organisation and experimental research design. Stylianou and Silver (2004) compared expert mathematicians' and undergraduate students' potential and actual use of visual representations in problem solving, by investigating their perceptions and analysing their methods. Similarly, Stepich and Ertmer (2009) conducted three related studies investigating novice-expert instructional designers' strategies for problem-finding, stating that the results of these studies can be used by educators to hasten the development of problem-finding expertise in their students. According to Fadde (2009), in the novice-expert studies, the important step is to identify "who is an expert?" Moreover, he advocates that "An obvious measure of expert status is amount of experience" (Fadde, 2009, p. 177). In the current study, which investigated the educational software evaluation processes of both experts and novices in conjunction with a software evaluation checklist developed by Heinich, Molenda, Russell and Smaldino (2002), the experts and novices were identified according to amount of experience.

Technology has become integral to instruction since the introduction of computers in the classroom. Kurz, Middleton and Yanik (2005) mention the promise of effective technology integration in school mathematics. Moreover, Virvou, Katsionis and Manos (2005) explain that students need to be motivated to learn during instruction, and that educational software can be both interesting and stimulating. Computer programs can

fulfill a variety of functions in the teaching and learning process, such as drill and practice, tutorials, instructional games, simulations, spreadsheets, word processing, database management and computer programming (De Corte, Verschaffel & Lowyck, 1994). However, as Niederhauser and Stoddart (2001) state, "Computer technology, in and of itself, does not embody a single pedagogical orientation" (p. 15). Well-designed review software offers technology while assuring predictable outcomes (Hooper & Hokanson, 2000, as cited in Kurz et al., 2005).

Yanpar Yelken (2011) further points out that selection of educational software is an important part of instruction. Similarly, Heinich et al. (2002) and Smaldino, Lowther and Russell (2008) explain the importance of media selection as part of a specific model, ASSURE: (a) analyse the learner, (b) state objectives, (c) select media, (d) utilise media, (e) require learner participation and (f) evaluate and revise. Many scholars, such as Heinich et al. (2002), De Villiers (2004), Niederhauser and Stoddart (2001), Cennamo, Ross and Ertmer (2010) and Squires and Preece (1996), provide checklists, evaluation forms and surveys to help educators select appropriate software for instructional purposes. To successfully integrate technology into their teaching, pre-service and novice teachers must gain expertise in the educational software selection process (Yanpar Yelken, 2011).

Practising is one of the important parts of the gaining expertise and transferring what is learned to a real context, like an expert (Sancar Tokmak & Karakus, 2011). Stepich and Ertmer (2009) criticise educational programs by stating that students often gain conceptual knowledge from educational programs, but do not apply that knowledge to problem-solving during practice. Further, Baumgartner and Payr (1996) advocate that the learner's goal should always be to become an expert in a field. Expertise based training (XBT) can be an alternative for instructors who aim to enable novice learners to gain expertise in a field and transfer knowledge to the job, since the focus of XBT, an instructional design theory, is to create strategies based on expert-novice studies to help learners develop advanced skills (Fadde, 2009). Chi, Glaser and Rees (1982) mention a relevant study: Chase and Simon investigated novice and expert chess strategies to examine the effects of knowledge on complex skill performance. Livingston and Borke (1990) emphasise the importance of expert-novice comparison research: "Research comparing how expert and novice mathematics teachers construct different types of lessons, manage content, and interact with their students may suggest ways to assist prospective and beginning mathematics teachers" (p. 372). Since expert-novices studies not only present the differences between two groups, but also includes clues to the key parts of expertise, novices can always get lessons from expert-novice studies to behave more like an expert.

Based on this literature review, the current novice-expert comparison study was conducted as a first step in enhancing students' expertise in educational software selection and evaluation. It was specifically designed to (a) compare the ways experts and novices evaluate educational software using the *Educational Software Evaluation Checklist* developed by Heinich et al. (2002) and (b) examine the implications of these variations in instructional practice.

Research questions

Two primary research questions drove this comparative case study, which investigated novice and expert selection processes for educational mathematics software:

1. How do novices and experts use the evaluation checklist to select educational software?
2. How compatible are the evaluations of novices and experts in terms of
 - a. the degree of consistency in scores; and
 - b. the percentage distribution of criteria in the evaluation checklist?

Method

This comparative case study used thematic comparisons. The researchers used themes and criteria from Heinich et al.'s (2002) *Educational Software Evaluation Checklist* to compare novice and expert software evaluation processes. Comparing two cases, in accordance with comparative case-study design, could help define the gap between novices' and experts' evaluation processes. The current study employed both qualitative and quantitative approaches. More specifically, data collection in the current study consisted of focus group interviews, classroom observations and document analysis.

Sampling

The researchers followed a specific sampling procedure with regard to research design, choosing the sampling purposively with regard to qualitative research, as prescribed by Patton (1990) and Miles and Huberman (1994). The participants of this study were both novices and experts; novices were defined as students who had not previously evaluated educational software. A total of 25 novice students enrolled in the course *Instructional Technology and Material Design* volunteered to participate and were placed in groups to evaluate educational software. One group (n=5 students) selected instructional chess software, while the other four groups (n=20 students) selected educational mathematics software. The group (n=5 students) that selected chess-education software was not taken into account, since the researchers applied a homogenous sampling strategy as a type of purposive sampling. Thus, the sampling of the study consisted of 20 students who selected educational mathematics software. Once the novices had selected their software, three experts were purposively selected, based solely on their fields of expertise: computer education and instructional technology, mathematics education and educational sciences, respectively.

Participants' background information

Data were collected from all participants. Before the sampling procedure, the novices had received a demographics questionnaire with questions about age, gender, department, enrolled courses, class level, GPA, home computer usage and coursework and other activities requiring computer usage. Twenty second-year college students (9 males, 11 females) from the night program in elementary education participated in the study as a part of the subject *Instructional Technology and Material Design*. Their ages ranged from 19 to 25, with an average of 21 (SD = 1.9). Their average GPA was 2.82 (out of 4.0) with an SD = .1. All had previously taken the subjects *Introduction to Education Sciences*, *Psychology of Education*, and *Teaching Technique and Methods*. Except for one student who did not have a computer, all students reported using computers actively, with experience ranging from 2 to 12 (average 6.3). Students had previously been required to take three subjects that necessitated using computers, during which they learned to use MS Office programs such as *PowerPoint*.

After the novices conducted their evaluations, an expert panel was chosen based on the topic of this study and the content of the software selected. All experts held doctoral degrees and had teaching experience. The expert from the field of computer education and instructional technology had been studying the development and usability of educational software for six years; the expert in educational sciences, for 14 years. The mathematics education expert focused on curriculum development and educational software use and evaluation.

Instrumentation

Six instruments were used to collect data in the current study, beginning with the demographics questionnaire described above, which was checked by a professor before being given out.

The second instrument was Heinich et al.'s (2002) *Educational Software Evaluation Checklist*. The checklist has three main parts: software description, rating and open-ended questions. Software description includes the title, serial title, keywords, format, source, date, cost, length, subject area, intended audience, brief description, objectives and entry capabilities required. The rating section has 11 points for evaluation: match with curriculum, accurate and current, clear and concise language, arouse motivation/maintain interest, learner participation, technical quality, evidence of effectiveness, free from objectionable bias, user guidance/documentation, clear directions, stimulates creativity and design principles, the last of which was added for course relevancy. The third part asks open-ended questions about the software's strengths and weaknesses, and asks respondents for recommendations for the software.

The third instrument was an observation form based on Heinich et al.'s (2002) checklist. It consisted of two parts: the criteria in the checklist and researcher notes on group comments made about the criteria during evaluation presentations.

The fourth instrument was the expert's *Educational Software Evaluation Checklist*, which included detailed explanations for each criterion in Heinich et al.'s (2002) checklist (see Appendix). The definitions were compiled using De Villiers (2004), Niederhauser and Stoddart (2001), Cennamo, Ross and Ertmer (2010) and Squires and Preece (1996). The three experts evaluated the selected educational software separately. During the evaluation process, each expert defined detailed criteria relevant to each main criterion in Heinich et al.'s (2002) checklist. The detailed criteria were analysed through inter-rater reliability, then discussed, and a unique criterion table was defined.

The fifth instrument was a focus group interview form prepared by the researchers and controlled by each expert professor. The form consisted of three questions:

1. How did group members evaluate the software (individually or together)?
2. To what extent did you benefit from the evaluation checklist?
3. How did you decide to grade the educational software as high, medium, or low?

The last instrument was an expert diary, in which the experts were requested to define their own evaluation process based on the evaluation checklist.

Procedure

This study followed several steps. First, the participants completed a demographic data questionnaire. Second, under the scope of the *Instructional Technology and Material Development* course, novice students discussed key issues in evaluating educational software. Third, the novices, in self-determined groups, selected and evaluated educational software using Heinich et al.'s (2002) checklist, presenting their findings in class. In the next step, the three experts evaluated the selected software using the same checklist. The final step was the analysis of the codes for both experts and novices.

Data analysis

The data analysis was descriptive, focusing on comparing how participants used the evaluation checklist, on the effects of experts' practical experiences on their evaluation processes, and on the compatibility between the experts' and novices' scores.

Open coding analysis was applied to observation notes from group presentations and the subsequent focus group interviews. Experts wrote diaries on their evaluation processes; these were similarly coded. Two researchers coded the data; according to Miles and Huberman's (1994) formula, interview intercoder reliability on the themes was 75%. After recognising and clarifying a disagreement on the name of one theme, intercoder reliability reached 100%.

Researchers organised the data collected from the experts into common themes, as outlined by Patton (1990). Moreover, as Ayres, Kavanaugh and Knafl (2003) recommend, each theme was described in detail, noting variations across themes. Then, the relationships between these themes were defined. This process was then applied to the novice groups' responses and observation notes. Lastly, expert and novice responses were compared to find similarities and differences in the educational software evaluation process with regard to the defined themes. Separate comparisons were first conducted within expert and novice groups, then expanded across both groups.

The software evaluation checklist results were compared descriptively. The experts evaluated the educational software that novices selected using Heinich et al.'s (2002) checklist separately, defining each criterion in detail. The coders' agreement rate on the criteria was 84%. After discussions on disputed items and corrections of miscalculations, coder agreement rate increased to 100%.

Validity issues

According to Maxwell (1996), validity is the goal of research studies, rather than a product, and a qualitative researcher should rely on common sense for credible conclusions. Research can be interpreted in many different ways, and the researcher should explain how he or she ruled out alternatives (Maxwell, 1996). This study's validity threats and coping strategies are given below:

- Sampling selection plays an important part in the validity of a study, as it can affect results. Expert and novice participants were identified based on their levels of knowledge of educational software evaluation. Before the study, a demographics questionnaire was presented to the novice students and background information

was collected about the experts' fields. There were no validity threats to the sampling selection in the current study.

- Instrumentation is another key piece of study validity, according to Wallen and Fraenkel (2001): a researcher cannot automatically assume instruments are valid. In the current study, all the developed instruments were verified by an expert before application.
- Data collection was triangulated by collecting data through focus group interviews, observations and document analysis.
- Peer debriefing was applied during data analysis. Two researchers analysed the data, and intercoder reliability was controlled via Miles and Huberman's (1994) formula.
- Data interpretation was discussed by the researchers. Moreover, for external conformability, an expert reviewed the study results.

Ethical issues

The researchers considered how the study could harm participants. The experts, who were from different fields, faced no potential harm. However, since the data were collected under the scope of a course where the instructor's attitude might have been affected by negative comments, all students who voluntarily participated in focus group interviews and demographic data collection were assigned pseudonyms. Since the software evaluation and presentation was a graded part of the course, the researchers acquired permission from participants before using the data.

Results

This comparative case study focused on novice and expert processes of evaluating educational software in the mathematics field using Heinich et al.'s (2002) *Educational Software Evaluation Checklist*. The subheadings below separate results according to the research questions of the study.

Differences in the use of the checklist by novices and experts

During the process of evaluating the software based on Heinich et al.'s *Educational Software Evaluation Checklist* (2002), experts and novices presented several differences. As a result of data analysis of observation notes taken during the novices' presentations, focus group interviews with novices and experts' diaries, four categories were created: focusing on the meaning of each criterion; searching for related literature; handling criteria; and grading policy.

At the beginning of the evaluation process, experts displayed detailed assessments of the meaning of each criterion. They aimed to construct a table for each criterion based on studies of software evaluation processes. One of the experts stated:

When I received the software evaluation checklist, I firstly thought about the meaning of each main criterion in the checklist. Then, I defined the detailed criteria in each main criterion in the list. I thought about each criterion separately.

The novice students did not indicate any complex efforts to understand the meaning of each criterion, leading to some misconceptions. For example, with respect to the criterion "stimulates creativity", novice students felt that creativity could only be stimulated by interesting and motivating activities such as games. However, the

criterion was actually related to providing opportunities for learner discovery and problem-solving. The same situation occurred with "evidence of effectiveness": although this criterion should be evaluated via students' test results, all groups measured it based on games, colouring, writing activities or stories in the software. During focus group interviews, no novice student said anything explicit about contemplating criterion meaning.

Another difference between expert and novice methods involved the consideration of literature and curriculum. For the "match with curriculum" criterion, the students only weighed whether the topics corresponded with the official curriculum, but ignored how the topic was treated in terms of curricular goals and skills to be acquired by learners. On the other hand, experts investigated the curriculum first, and only then evaluated whether the software was suitable for each curricular goal.

The two groups' methods also varied in terms of handling criteria. The experts defined detailed criteria and created a primary checklist, evaluating the software from many sides. However, the novices identified only one main property. For example, the inclusion of a game earned a high grade for "arouse motivation/maintain interest", without an assessment of variety and quality. For "student participation", interaction through answering questions was marked as high by the novices, while the experts checked the software in terms of learner control, interactivity, maintenance of curiosity and confidence, effectiveness of feedback and means of participation.

The grading policy presented a fourth difference between the experts and novices. The experts' diaries showed that they created a detailed scale based on Heinich et al.'s (2002) checklist, incorporating the same measures for each criterion to increase consistency and reliability. They assigned each criterion a positive or negative sign with respect to software properties. In evaluating design principles, for example, the experts defined 12 detailed sub-points that they evaluated for each software program as negative, positive, or neutral (see Appendix). To categorise "design principles" as high, medium, or low, the experts counted the numbers of negative, positive and middle signs for the detailed criteria. One expert explained her grading policy:

While grading the software, I defined detailed criteria for each main criterion in Heinich et al.'s (2002) checklist. I checked whether it had the necessary properties with respect to the each detailed criterion. Then I marked each criterion as positive or negative with respect to the software properties. If the negative signs were 25% or less I signed the main criterion as positive; if the positive signs were 25% or less I signed the main criterion as negative. If the negative and positive signs were nearly equal, I signed the main criterion as neutral.

However, observation notes showed that the novices did not use any rubric during the process. They graded each criterion as high, medium or low, but they did not provide details about specific properties in their evaluation checklist or during their presentations. Moreover, in focus group interviews, all novice groups stated that they evaluated the software according to general impressions. For example, while evaluating design principles, the novice groups focused on "consistency of the colours, texts or objects" and "use of contrast colours between letters/ pictures/ elements and background in the software". They ignored important criteria such as "avoiding excessive use of highlighting techniques on letters, pictures, animation", "avoiding excessive use of texts, pictures, animations, sounds, etc.", "appropriateness of computer-screen layout and ease of reading/ following", "appropriate use of margins for consistency" and "indication of completed sections of the software".

It is important to emphasise the differences between experts' and novices' definitions of two more elements of the evaluation checklist: cost-effectiveness and target audience. When judging cost-effectiveness, three out of four novice groups presented a general misunderstanding that cost-effectiveness is based solely on literal cost: if something is free, it automatically qualifies as cost-effective. They tended to ignore the fact that effectiveness must also measure the degree of fulfilment of targeted goals. Moreover, three out of four novice groups presented a poor match between the content of the software and appropriate target audience, as they were not aware of the benefits of considering the literature.

Compatibility between novices' and experts' scores using the checklist

Novice and expert scores on Heinich et al.'s (2002) checklist were analysed descriptively to assess score compatibility in terms of the degree of consistency and percentage distribution of standards. Table 1 gives the agreement rates of scores based on the software evaluation checklist. The general agreement rate for all software was 48%. For "clear and concise language", participants showed 100% agreement, whereas they presented no agreement on "stimulates creativity".

Table 1: Novice-expert agreement rates on software evaluation

Criteria	Agreement	Disagreement	Agreement rate (%)
Match with curriculum	1	3	25
Accurate and current	3	1	75
Clear and concise language	4	0	100
Arouse motivation/maintain interest	2	2	50
Learner participation	1	3	25
Technical quality	3	1	75
Evidence of effectiveness	1	3	25
Free from objectionable bias	3	1	75
User guidance/documentation	3	1	75
Clear directions	1	3	25
Stimulates creativity	0	4	0
Design principles	1	3	25
Total	23	25	48

As shown in Table 1, the compatibility rate between novice and expert scores in terms of consistency was high for five criteria: "accurate and current", "clear and concise language", "technical quality", "free from objectionable bias" and "user guidance/documentation". For agreed items, respondents rated 74% of criteria as "high", whereas they rated 22% as "medium" and 4% as "low".

Table 2 gives the distribution of scores for four educational software programs, as indicated by the evaluation checklist. In general, students marked the software as highly compatible with each criterion, rating 75% of the criteria as high, 23% as medium and 2% as low. For all software, students marked "match with curriculum", "accurate and current" and "free from objectionable bias" as high. Only one group of students ranked one software product as low, because it did not provide any guidance or a manual. On the other hand, experts did not put any software on a single level. They differed from the novice students by giving more low marks under "match with curriculum" and "evidence of effectiveness". Therefore, students presented weaknesses in knowing the content and skills presented in the related curriculum, and did not demonstrate a clear understanding of the criteria in the evaluation checklist.

Table 2: Distribution of novice-expert evaluation scores based on levels

Criterion	Novice			Expert		
	High	Medium	Low	High	Medium	Low
Match with curriculum	4	0	0	1	1	2
Accurate and current	4	0	0	3	1	0
Clear and concise language	3	1	0	3	1	0
Arouse motivation/ maintain interest	4	0	0	2	2	0
Learner participation	3	1	0	1	3	0
Technical quality	3	2	0	1	3	0
Evidence of effectiveness	1	2	0	1	1	2
Free from objectionable bias	4	0	0	3	1	0
User guidance/ documentation	2	1	1	0	2	2
Clear directions	3	1	0	2	1	1
Stimulates creativity	2	1	0	0	2	2
Design principles	2	2	0	3	1	0
Total	35	11	1	20	19	9
%	75	23	2	41	40	19

Although experts tended to rate criteria as "high" more often, at 41%, their rates were more closely distributed across the levels (Figure 1). In general, 48% of novice-expert scores were in agreement. For the novice students, 46% were higher, whereas 6% were lower. Clearly, novice students tended to grade more highly than experts.

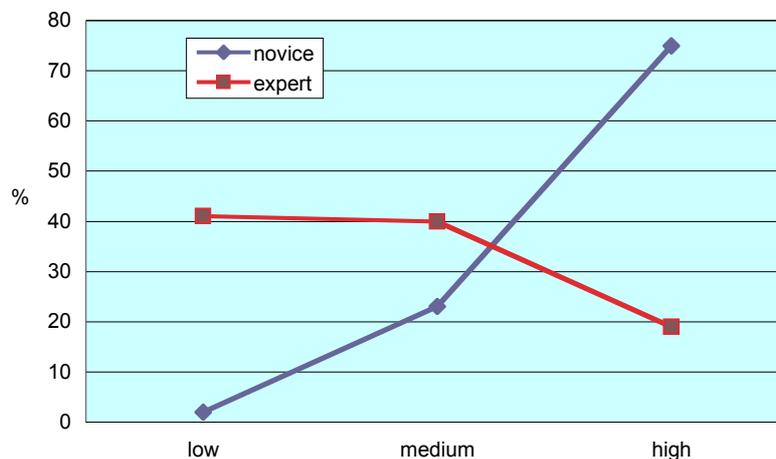


Figure 1: Percent distribution of novice and expert scoring

Discussion and conclusion

Among the main findings of the study was that while using Heinich et al.'s (2002) checklist, novice students presented weaknesses in understanding the meaning of each main criterion, supporting the process with related literature, handling each criterion and grading. The experts tended to weigh the meaning of each criterion and search the literature for further understanding, whereas novices simply discussed the criteria and started to assess. Similar results were found by Swanson, O'Connor and Cooney (1990) in a study that aimed to investigate differences between expert and novice teachers in solving classroom discipline problems. The expert teachers identified priorities while

defining the problem, but novice teachers tended to represent problems strictly in terms of possible solutions (Swanson, O'Connor & Cooney, 1990). Moreover, in Stepich and Ertmer's 2009 study, which investigated ill-structured problem-solving strategies of novices and experts, the experts spent more time problem-finding and relating details of the main issues. Similarly, in the current study, the experts defined detailed standards for each main criterion. Figure 2 shows the differences between experts and novices during the software evaluation process. Experts' steps began with individuals searching the literature to improve their understanding of the meaning of each criterion, coming together to discuss and clarify their understanding, developing a detailed criterion list to handle each main criterion and grading educational software based on the number of detailed criteria they developed. On the other hand, novices' process of software evaluation included first having a discussion about how to handle each criterion, then grading the software according to their impressions.

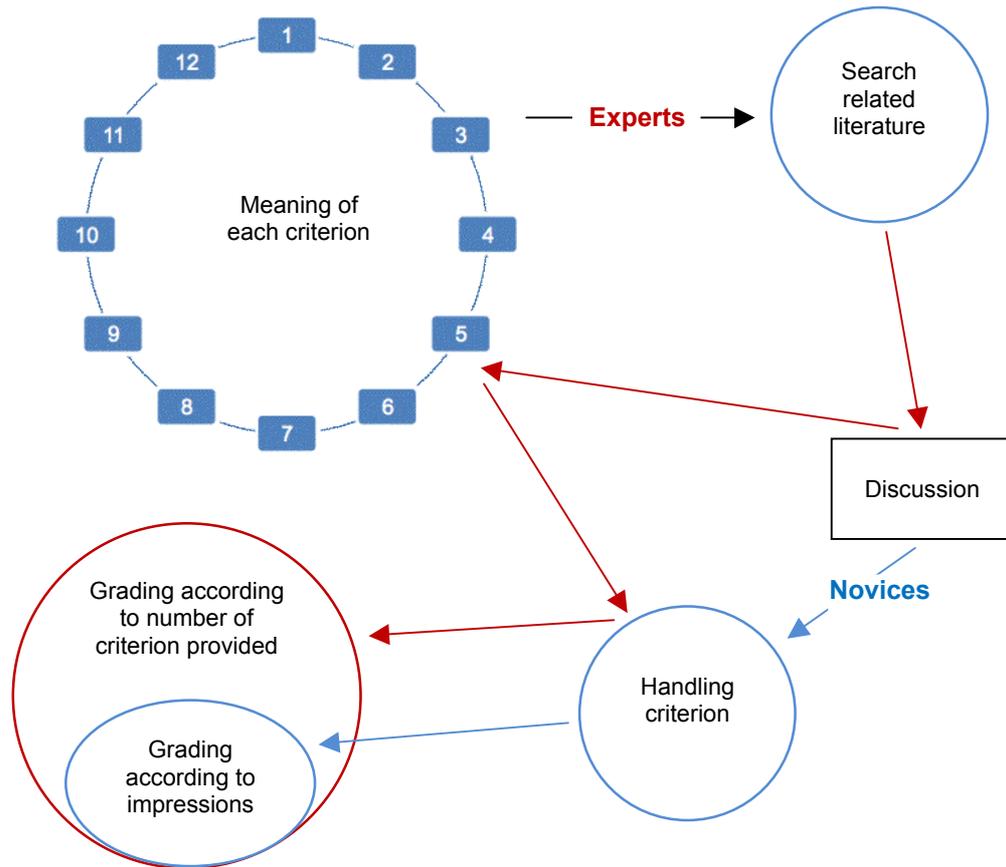


Figure 2: Differences between experts and novices during the software evaluation process

The expert-novice agreement rate was as low as 48%, and mostly based on highly rated items: novices generally graded each criterion higher than the experts. Meyer (2004) explains one of her study's findings about the difference between novice and expert teachers' conceptions of prior knowledge by pointing to "an apparent mismatch

between the novice teachers' beliefs about their urban students' life experiences and prior knowledge and the wealth of knowledge the expert teachers found to draw upon" (p. 970). In the current study, differences may have been caused by students misinterpreting criteria, by the limited methods for evaluating each criterion, by students' sparse of knowledge of content and skills addressed in the curriculum and by the lack of a common grading strategy.

The findings of the current study have many implications for understanding the gap between novice and expert, with regard to software evaluation. Educators can fill this gap and hasten the development of pre-service or new teachers by taking into account the process differences identified in this study. As Stepich and Ertmer (2009) state, expertise can be gained via practice. For that reason, more students should be given the opportunity to be involved in software evaluation, selection and development.

This study will help future researchers understand how experts and novices act differently during the process of educational software evaluation. In an earlier study, Kozma and Russell (1997) investigated what could be done to bridge the gap between student and expert thinking in chemistry curriculum, instruction and assessment. The results of this study are also beneficial for researchers in the field of mathematics education who want to investigate how to close the gap between novice and expert, with regard to software evaluation. Additional research is needed to investigate how the results of this study would differ if novices had access to the detailed criteria table created by the experts and were provided with general grading guidelines.

References

- Ayres, L., Kavanaugh, K. & Knafl, K. A. (2003). Within-case and across-case approaches to qualitative data analysis. *Qualitative Health Research*, 13(6), 871-883. <http://dx.doi.org/10.1177/1049732303013006008>
- Baumgartner, P. & Payr, S. (1996). Learning as action: A social science approach to the evaluation of interactive media. In P. Carslon & F. Makedon (Eds.), *Proceedings of ED-MEDIA 96, World Conference on Educational Multimedia and Hypermedia*, Charlottesville, VA, 31-37.
- Cennamo, K., Ross, J. & Ertmer, P. (2010). *Technology integration for meaningful classroom use: A standards-based approach*. Belmont, CA: Wadsworth, Cengage Learning.
- Chi, M. T. H., Feltovich, P. J. & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5(2), 121-152. http://dx.doi.org/10.1207/s15516709cog0502_2
- Chi, M. T. H., Glaser, R. & Rees, E. (1982). Expertise in problem solving. In R. J. Sternberg (Ed.), *Advances in the Psychology of Human Intelligence* (Vol. 1). Hillsdale, NJ: Erlbaum.
- Crossley, M. & Vulliamy, G. (1984). Case-study research methods and comparative education. *Comparative Education*, 20(2), 193-207. <http://www.jstor.org/stable/3098564>
- De Corte, E., Verschaffel, L. & Lowyck, J. (1994). Computers and learning. In T. Husen & T. N. Postlethwaite (Eds.), *The international encyclopedia of education* (2nd ed.). Elsevier Science.
- De Villiers, R. (2004). Usability evaluation of an e-learning tutorial: Criteria, questions and case study. In G. Marsden, P. Kotze & A. Adesina (Eds.), *Proceedings of the 2004 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries* (pp. 284-291). <http://dl.acm.org/citation.cfm?id=1035092>

- Fadde, P. J. (2009). Expertise-based training: Getting more learners over the bar in less time. *Technology, Instruction, Cognition and Learning*, 7(2), 171-197. [verified 10 Sep 2012]
<http://peterfadde.com/Research/xbttraining.pdf>
- Heinich, R., Molenda, M., Russell, J. D. & Smaldino, S. E. (2002). *Instructional media and technologies for learning* (7th ed.). Upper Saddle River, NJ: Prentice-Hall.
- Kurz, T. L., Middleton, J. A. & Yanik, H. B. (2005). A taxonomy of software for mathematics instruction. *Contemporary Issues in Technology and Teacher Education*, 5(2), 123-137.
<http://www.citejournal.org/articles/v5i2mathematics1.pdf>
- Kozma, R. & Russell, J. (1997). Multimedia and understanding: Expert and novice responses to different representations of chemical phenomena. *Journal of Research in Science Teaching*, 34(9), 949-968. [http://dx.doi.org/10.1002/\(SICI\)1098-2736\(199711\)34:9<949::AID-TEA7>3.0.CO;2-U](http://dx.doi.org/10.1002/(SICI)1098-2736(199711)34:9<949::AID-TEA7>3.0.CO;2-U)
- Livingston, C. & Borko, H. (1990). High school mathematics review lessons: Expert-novice distinctions. *Journal for Research in Mathematics Education*, 21(5), 372-387.
<http://www.jstor.org/stable/749395>
- Maxwell, J. A. (1996). *Qualitative research design: An integrative approach*. Thousand Oaks, CA: Sage.
- Meyer, H. (2004). Novice and expert teachers' conceptions of learners' prior knowledge. *Science Education*, 88(6), 970-983. <http://dx.doi.org/10.1002/sce.20006>
- Miles, M. B. & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Thousand Oaks, CA: Sage.
- Niederhauser, D. S. & Stoddart, T. (2001). Teachers' instructional perspectives and use of educational software. *Teaching and Teacher Education*, 17(1), 15-31.
[http://dx.doi.org/10.1016/S0742-051X\(00\)00036-6](http://dx.doi.org/10.1016/S0742-051X(00)00036-6)
- Patton, M. Q. (1990). *Qualitative research and evaluation methods* (2nd ed.). Newbury Park, CA: Sage.
- Reiser, R. A. & Kegelmann, H. W. (1994). Evaluating instructional software: A review and critique of current methods. *Educational Technology Research and Development*, 42(3), 63-69.
<http://dx.doi.org/10.1007/BF02298095>
- Sancar Tokmak, H. & Karakus, T. (2011). ICT pre-service teachers' opinions about the contribution of initial teacher training to teaching practice. *Contemporary Educational Technology*, 2(4), 319-332. <http://www.cedtech.net/articles/24/245.pdf>
- Smaldino, S. E., Lowther, D. L. & Russell, J. D. (2008). *Instructional technology and media for learning* (9th ed.). Upper Saddle River, NJ: Pearson Merrill.
- Squires, D. & Preece, J. (1996). Usability and learning: Evaluating the potential of educational software. *Computers & Education*, 27(1), 15-22. [http://dx.doi.org/10.1016/0360-1315\(96\)00010-3](http://dx.doi.org/10.1016/0360-1315(96)00010-3)
- Stepich, D. A. & Ertmer, P. A. (2009). Teaching instructional design expertise: Strategies to support students' problem-finding skills. *Technology, Instruction, Cognition and Learning*, 7, 147-170.

Stylianou, D. A. & Silver, E. A. (2004). The role of visual representations in advanced mathematical problem solving: An examination of expert-novice similarities and differences. *Mathematical Thinking and Learning*, 6(4), 353-387. http://dx.doi.org/10.1207/s15327833mtl0604_1

Swanson, H. L., O'Connor, J. E. & Cooney, J. B. (1990). An information processing analysis of expert and novice teachers' problem solving. *American Educational Research Journal*, 27(3), 533-556. <http://dx.doi.org/10.3102/00028312027003533>

Virvou, M., Katsionis, G. & Manos, K. (2005). Combining software games with education: Evaluation of its educational effectiveness. *Educational Technology & Society*, 8(2), 54-65. http://www.ifets.info/journals/8_2/5.pdf

Wallen, N. E. & Fraenkel, J. R. (2001). *Educational research: A guide to the process* (2nd ed.). Mahwah, NJ: Erlbaum.

Yanpar Yelken, T. (2011). *Ogretim teknolojileri ve material tasarımı* (2nd ed.). Ankara: Ani Yayıncılık.

Appendix: Experts' educational software evaluation checklist

Criteria	1. Match with curriculum			2. Accurate and current			3. Clear and concise language			4. Arouse motivation/maintain interest		
Detailed criteria	a. Meeting the curriculum goals b. Appropriateness of curriculum philosophy <ul style="list-style-type: none"> • Critical thinking • Creative thinking • Communication • Decision-making • Problem-solving • Exploring • Entrepreneurship 			a. Accuracy of definitions b. Completeness of definitions c. Currency of definitions d. Accuracy of notations e. Completeness of notations f. Currency of notations g. Accuracy of symbols h. Completeness of symbols i. Currency of symbols			a. Appropriateness to target learners' reading level b. Sufficiency of character sizes c. Understandability of texts/speeches, etc. d. Avoidance of jargon e. Consistency in clear use of fonts			a. Learner control b. Variety of interactivity c. Maintenance of curiosity and confidence d. Effectiveness of feedback to students' responses <ul style="list-style-type: none"> • Positive and negative feedback • Correctness of feedback • Indication of students' correct response • Re-teaching the lesson or concept after multiple incorrect attempts • Student performance report (print out or on-screen) e. Engagement with daily life <ul style="list-style-type: none"> • Through objects/stories/cartoons • Through activities such as problem-solving, creating characters f. Appropriateness to learners' age/gender etc.		
Rating	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()

Criteria	1. Learner participation			2. Technical quality			3. Evidence of effectiveness			4. Free from objectionable bias		
Detailed criteria	a. Requiring active participation of target users b. Variety of participation <ul style="list-style-type: none"> • Selection • Entering answers • Matching • Repeating through writing or spelling 			a. Proper functioning of all links b. Proper functioning of all multimedia resources at all times c. Ease of installing software d. Availability of clear directions for access or installation e. Use with peripherals f. Booting of program g. Adjustable sound level h. Minimum level of technical expertise to run program i. Cost-effectiveness j. Program operation without crashing k. Multiple means to operate program l. Terminating the program from any place at any time			a. Test results b. Ongoing monitoring of target learners' progress c. Variety of assessments			a. Avoiding stereotypes b. Avoiding biases (gender, race, culture, etc.)		
Rating	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()

Criteria	1. User guidance/ documentation	2. Clear directions	3. Stimulates creativity	4. Design principles
Detailed criteria	a. Completeness of user guidance/ documentation b. Correctness of user guidance/ documentation c. Understandability of user guidance/ documentation d. Currency of user guidance/ documentation e. Accessibility of user guidance/ documentation from anywhere in software	a. Correctness of grammar and spelling b. Appropriateness of directions to target learners' level c. Sufficiency of directions d. Evidence of functions of navigational elements/icons	a. Existence of discovery b. Discovery through automatic guidance c. Discovery through optional guidance	a. Consistency in letters, colours, buttons, pictures, etc. used b. Avoiding excessive use of highlighting techniques on letters, pictures, animation c. Avoiding excessive use of texts, pictures, animations, sounds, etc. d. Use of contrasting colours between letters/ pictures/ elements and background e. Appropriate spacing between elements f. Appropriateness of size of letter/ picture/ animation etc. to target learners' age g. Appropriateness of computer-screen layout and ease of reading/ following h. Appropriate use of margins for consistency

													i. Appropriate use of chunks j. Use of same colours for elements in the software as ones in real life k. Equal distribution of elements on screen l. Indication of completed sections of the software m. Providing more than one way of navigating through the content
Rating	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	Low ()	Medium ()	High ()	

Authors: Assistant Professor Hatice Sancar Tokmak
 Computer Education and Instructional Technology Department
 Mersin University, Turkey
 Email: haticesancarr@gmail.com Web: <http://haticetokmak.wordpress.com/>

Assistant Professor Lutfi Incikabi
 Mathematics Education Department, Mersin University, Turkey
 Email: lutfiincikabi@yahoo.com

Professor Tugba Yanpar Yelken
 Educational Sciences Department, Mersin University, Turkey
 Email: tyanpar@gmail.com Web: <http://www.mersin.edu.tr/apbs/tyanpar>

Please cite as: Sancar Tokmak, H., Incikabi, L. & Yanpar Yelken, T. (2012). Differences in the educational software evaluation process for experts and novice students. *Australasian Journal of Educational Technology*, 28(8), 1283-1297.
<http://www.ascilite.org.au/ajet/ajet28/sancar-tokmak.html>