Research Report

Mark Rash

markrash@vt.edu

August 1, 2010

Research Report

Problem Statement

With so many software applications becoming an everyday part of life for today's workers, and with limited time and facilitator resources for training employees on those applications, there is a need to determine best practices for effectively and efficiently training employees on using software on the job. The focus of exploration on this topic is on ways in which Instructional Technology can help establish best practices in the area of software training. The focus is not on Instructional Technology in terms of suggesting particular media or technology to solve the problem, since repeated studies have shown that no certain medium in and of itself enhances learning more than other media, regardless of all other considerations (Clark & Salomon, 1986, p.13). Instead, the focus of the topic is on proven best practices for using Instructional Technology – which encompasses much more than media, but also instructional design and all other aspects of teaching and learning – to successfully and efficiently train end users on specific software applications. Groups associated with the topic include corporate trainers, software end users, and instructional technologists.

Preliminary Research Question

What instructional approaches have been proven effective in teaching users to correctly use software applications?

Literature Review

In the interest of exploring the preliminary research question, the author selected a number of relevant articles from primary sources. All of the literature reviewed for this purpose encompassed research studies that cover various aspects of instructional methodologies used in teaching software to learners. The following section discusses some of the research that has been

conducted in response to the preliminary research question, as well as practical implications of the findings.

Harp, Taylor, and Satzinger conducted research on three commonly-employed methods for training software: computer-based training (CBT), instructor-led training, and video training (1998, p.271). The intent of the study was to determine the effectiveness of these three approaches from the perspective of the learners. The researchers found that the perceived effectiveness of software training was heavily influenced by prior learner experience and overall comfort with computers, as well as learners' general tendency toward either dependent or self-directed learning (Harp, Taylor, & Satzinger, 1998, p.281). The implication is that careful analysis of the target audience must yield information as to the level of field-dependence or field-independence exhibited by the learners in terms of preferring dependent or self-directed learning. If the target audience tends to prefer self-directed learning, CBT can be an effective substitute for instructor-led training when accompanied by a meaningful instructional strategy (Harp, Taylor, & Satzinger, 1998, p.281). On the other hand, learners with a preference for more mediated instruction will need a facilitated session in order to learn the software most effectively.

One of the approaches to software training used by the author's current employer involves the use of a "training wheel," which is a version of the software being trained that has irrelevant functions blocked, disabled, or removed during training to reduce demands on learners' cognitive capacity (Bannert, 2000, p.337). Bannert conducted research to evaluate the effectiveness and efficiency of two variations of learner support materials during software training: the limitation of functions available within the user interface during training and the use of self-directed support materials during training (2000, p.336). When measuring learner

satisfaction, the study found that learners perceived courses that used software training wheels to be less effective than those using the complete software interface (Bannert, 2000, p.341). In confirmation of countless other research that has been done in various settings, instructional media (in this case, the use of self-directed support materials) was found to have no significant impact on learning outcomes (Bannert, 2000, p.341). However, learning efficiency, as measured by the time needed for learning in the experimental treatments, was most affected by instructional method, with self-learning material being completed about two hours faster than the traditional approach (Bannert, 2000, p.340). The 'training wheels' approach led to more efficient learning in terms of the time required to complete the materials; but neither approach resulted in a significant difference in learning outcomes (Bannert, 2000, p.342). The implication is that the use of training wheels in software training is really only recommended when time schedules are tight, since the participants were found to be less acceptant of the limited interface in the learning environment when compared to the performance environment (Bannert, 2000, p.343).

Another consideration in terms of providing learner support during software instruction centers on the type of application help provided to the user. Dutke & Reimer conducted research to evaluate two forms of software help as a support mechanism for learners: operative help, or a list of action steps that must be taken to achieve a goal, and function-oriented help, or an explanation of how a particular software function works (Dutke & Reimer, 2000, p.307). The research involved two experiments, the second of which introduced a change in schema to reflect a realistic change likely to be encountered when using the software to perform specific tasks. Function-oriented help was determined to better-support learners in the schema change than operative help (Dutke & Reimer, 2000, p.313). For both experiments, function-oriented help promoted better skill transfer than operative help, but the latter resulted in greater learning

efficiency prior to the schema change (Dutke & Reimer, 2000, p.313). The practical implication for software training is that tasks intended to consolidate acquired action-oriented knowledge structures should include operative online help (Dutke & Reimer, 2000, p.313). Tasks intended to keep learners within the bounds of existing schemas should include function-oriented help as a support mechanism (Dutke & Reimer, 2000, p.313).

Andrew Hurt, of Texas A&M University, conducted a study of ways that software trainers apply theories of andragogy (i.e. adult learning), situated cognition, and minimalist methodology to software training (Hurt, 2007). Essentially, Hurt found that androgogy was the key mediating variable across the entire process, with principles of adult learning being forefront in the mind of the software trainers studied (Hurt, 2007). Additionally, while Hurt cited prior research that found that software trainers should use either systematic or minimalist methods for software instruction – but not both (Lambrecht, 1999, as cited in Hurt, 2007) – general practice seems to include systematic training for the basic software functions and a minimalist approach for further clarification and specific problem-solving skills (Hurt, 2007). The implication for software training is that adult learning principles should be used to develop the instructional strategy when designing software training and that the strategy should employ either minimalist or systematic instruction, depending on the complexities of the software and related tasks, and with consideration of learner preferences and styles.

A study by Shailendra Palvia and Prashant Palvia explored the effectiveness of using computers in software training (2007, p.479). The study was not on the mere selection of computers as a medium for instruction, but on a complete instructional strategy, with particular attention to learner satisfaction (Palvia & Palvia, 2007, p.479). The results found that learner satisfaction in software training was directly impacted by the use of computers in software

instruction (Palvia & Palvia, 2007, p.487). While, as with the other studies, the introduction of computers did not result in higher achievement, learners were more satisfied and confident in their abilities to transfer their knowledge and skills when they were able to practice concurrently on computers (Palvia & Palvia, 2007, p.487). Since learner attitudes and perceptions have a great deal of impact on ultimate behaviors in the performance context, the research implies that providing concurrent practice on computers during software training may indirectly contribute to improved performance by way of impacting learner self-efficacy.

Much of the research substantiates the need for further investigation. Bannert suggests that additional study should be done on ‘training wheels,’ altering the implementation to determine of other variations could lead to greater learner acceptance of the training environment (2000, p.343). Harp, Taylor, and Satzinger suggest that additional research should be done to determine if other learner differences impact learner preference and perceived utility of software training conducted via CBT, video training, or instructor-led training (1998, p.281). Additionally, Hurt recommends future research to explore the relationship that he inadvertently discovered between the theories of andragogy, minimalist instruction, and situated cognition as they apply to software training, as the implications of the relationship is unclear (Hurt, 2007). The author would agree with this recommendation, considering that strong research often involves investigating relationships (Fraenkel & Wallen, 2009, p.34). Finally, Palvia and Palvia suggest that researchers use their evaluation framework to further define how, when, and why computer-based instructional methods should be used in software instruction (Palvia & Palvia, 2007, p.488).

Summary and Conclusions

The research results presented in the literature review support the research question, in that they discuss specific recommendations with regard to instructional approaches that have been found effective in software training. The literature explores the research question from a few different angles, including specific methodologies, theoretical frameworks, and special considerations that need to be made for adult learners in light of their existing knowledge and life experiences. The research focuses less on media selection and more on effective approaches to training learners to use software correctly.

As the author expected, the literature review exposed a variety of established best practices and recommendations for software training. However, a holistic view of the findings results in some important concepts to consider in the design, delivery, and implementation of software training. First, whether or not to employ computer-based training for software instruction should be primarily based on the target audience's preferences in terms of self-guided or mediated learning. If self-directed learning is preferred, computer-based training can be effective; a preference for mediated instruction is best suited with facilitator-led software training. Second, the 'training wheels' approach to software training is really only recommended when time schedules are tightly constrained, due to the lower user acceptance of the limited interface in the learning environment when compared to the performance context. Third, determining the appropriate learner support mechanisms to use during software training should be based on whether the task being trained is intended to build new knowledge structures (in which case operative online help is recommended) or to reach the bounds of existing schemas (in which case functional help is needed).

An important theme seems to prevail through the reviewed literature: principles of adult learning should be key drivers in software training. This is explicitly reflected in the research involving andragogy and its implications for software training, but is also implicitly reflected in the findings that suggest that learner preferences and perceptions will directly influence the effectiveness of software training and, thus, should impact the design and development of such instruction. Since adults are generally autonomous and self-directed (Lieb, 1991), it is important to include self-guided learning components in software instruction – and this was confirmed by some of the aforementioned conclusions based on the reviewed literature. Clearly, as with all other types of instruction, learner preferences need to be taken into account when developing an instructional strategy for software training, as these preferences have a direct impact on defining what would be considered the 'best practice' for a particular software training scenario.

Revised Research Question

Since the preliminary research question does not focus on what specific *media* results in the most effective software training, the author does not believe the question needs to be modified. The question instead focuses on specific strategies and methods that lead to effective results in software training. The literature review addressed the question as proposed by listing the key factors that determine success in software training.

References

Bannert, M. (2000). The effects of training wheels and self-learning materials in software training. *Journal of Computer Assisted Learning*, *16*(4), 336-346. doi:10.1046/j.1365-2729.2000.00146.x.

Clark, R. E., & Salomon, G. (1986). Media in teaching. *Handbook of research on teaching* (1-15).

Dutke, S., & Reimer, T. (2000). Evaluation of Two Types of Online Help for Application Software. *Journal of Computer Assisted Learning*, *16*(4), 307-15. Retrieved from ERIC database

Fraenkel, J. R., & Wallen, N. E. (2009). *How to design and evaluate research in education* (M. Ryan & D. S. Patterson, Eds., 7th ed.).

Harp, C., Taylor, S., & Satzinger, J. (1998). Computer Training and Individual Differences: When Method Matters. *Human Resource Development Quarterly*, *9*(3), 271-283. Retrieved from Education Research Complete database.

Hurt, A. (2007). Exploring the Process of Adult Computer Software Training Using Andragogy, Situated Cognition, and a Minimalist Approach. *Online Submission*, Retrieved from ERIC database.

Lieb, S. (1991). *Principles of adult learning.* Retrieved August 1, 2010, from http://honolulu.hawaii.edu/intranet/committees/FacDevCom/guidebk/teachtip/adults-2.htm

Palvia, S., & Palvia, P. (2007). The Effectiveness of Using Computers for Software Training: An Exploratory Study. *Journal of Information Systems Education*, *18*(4), 479-489. Retrieved from ERIC database.