

Stop blaming the teachers: The role of usability testing in bridging the gap between educators and technology

Jay Buzhardt and Linda Heitzman-Powell
University of Kansas

Abstract

Despite the often reported benefits of educational technology, educators often find it difficult to integrate these applications and devices into typical school practices. Although there are a number of complex factors and interactions that contribute to this problem, the usability of educational technology is rarely considered. The current paper discusses the role of 'usability testing' to improve technology integration into classroom practices, including common usability testing methods and measures. Finally, we discuss how educators and administrators can influence developers to improve their usability testing and their reporting of those tests.

The marriage between education and technology has often been rocky. It sometimes feels more like an 'arranged marriage' than a natural convergence of two seemingly compatible fields. Historically, the federal government has tried to improve this relationship by adding technology initiatives to its educational programs, including President Clinton's "National Call to Action" to connect all public schools to the Internet (Office of the Press Secretary, 2000), and President Bush's No Child Left Behind initiative (US Dept of Education, 2002), which requires all students to be 'technologically literate' by eighth grade. These initiatives have led to an increase in access to technology and spawned local legislation to further improve technology integration in specific states. For example, by 2003, 100% of all U.S. public schools had access to the Internet, and at least 37 states required teachers to receive technology training and/or demonstrate some level of technology proficiency (National Center for Education Statistics, 2005; Park & Staresina, 2004). However, these data can be deceiving because increased technology *access* and/or *knowledge* of technology historically has had little effect on classroom practice (Bottino, Forcheri, & Molfino, 1998). For example, the 2003 National Assessment of

Educational Progress conducted a national survey of fourth grade math teachers' use of computers in the classroom. Based on this survey, 40% of teachers reported that they primarily used computers for math games, 28% did not use computers at all, and 24% used them for "drill & practice". Also, computer and Internet use continues to be much less prevalent in schools that serve a significant percentage of Title I students and those in high-poverty areas (Coley, Cradler, & Engel, 1998; Swain & Pearson, 2002). Furthermore, Cuban (2001) notes that only 20% of teachers report that computers have significantly changed their classroom practices. In other words, schools' and classrooms' use of technology may appear to be increasing, but actual classroom instruction is limited at best.

Ultimately, the burden of bridging this gap between technology and teachers is placed squarely in the laps of teachers. They face the daunting task of not only using the technology, but also showing the expected benefits of its use (e.g., improved student outcomes, more efficient classroom management, reduced paperwork, etc.). Thus, teachers "fear of technology" or lack of technological expertise is often linked to teachers' use of technology in their classroom/instructional practices (Ike, 1997; Stone, 1998). Another barrier often cited is the contextual restraints of school settings which tend to hinder the implementation of *any* significant change (Bottino, Forcheri, & Molfino, 1998, Cuban, 2001). However, we rarely look to the specific technology itself and its usability as contributing to the lack of technology integration in classroom practices and instruction.

The current paper asserts that an infinite amount of training and improvement to schools' ability to implement change will do little to overcome the poor usability of a software application or website. Although technology developers are responsible for optimizing the usability of their products, the education community (e.g., teachers, administrators, district superintendents,

legislators, etc.) must be responsible purchasers and consumers of educational technology. This community has a right to expect and demand technology that has undergone rigorous usability testing and has data to support its use in educational settings. Currently, there is a movement within the technology industry to require technology developers to make standardized usability data available to consumers (Thibodeau, 2002). Although it is unlikely that the government would step in to enforce such a requirement, pressure from consumers (particularly large corporations) has forced some developers to conduct more rigorous usability tests and to make the findings available to potential purchasers. As a major consumer of software applications and web services, educators should join this movement by taking a stronger interest in learning about the usability of the technology they purchase. Including “evidence of usability” as a criterion for purchasing technology removes some of the barriers to using technology, and encourages developers to make usability testing a larger part of their R&D efforts. The current paper 1) provides a brief overview of the methods and measures used in rigorous usability evaluations, 2) discusses ways to assess usability in the absence of formal usability testing reports, and 3) explores how an educational community that is informed about usability can encourage the development of more usable educational technology.

What is Usability and How Is It Tested?

Jakob Nielsen, a leader in the field of Human-Computer Interaction (HCI), asserts that a system’s usability is determined by the degree to which it is “easy to learn, efficient to use, easy to remember, subjectively pleasing, [and causes] few errors” (Nielsen, 1993, p. 25). The popularity of technology intended for public use is often directly related to its usability. Two recent examples of usability’s impact on product use are Apple’s iPod, the dominant MP3 player on the market today, and the Google search engine. Reviews of the iPod suggest that its

popularity and appeal are primarily due to its usability, despite its high cost and somewhat limited functionality relative to other players (Machrone, 2004). While the popularity of Google is largely based on its superior functionality relative to other search engines, the simplicity of its interface (a single field and a “search” button) makes it an attractive application for even the most novice Internet user. In fact, since the rise of Google, other search engines have mirrored its simple interface (e.g., AltaVista, A9, AskJeeves, etc.)

Although usability has always been an issue in the computer industry, it was not until the mid-1980s that *usability testing* gained favor among producers of personal computers and software applications. At this time, the union of the ‘mouse’ (developed by Xerox in 1970) with the Macintosh graphical user interface (GUI – pronounced “goeey”) opened the computer world to the general public, going beyond ‘techie’ and highly skilled programmers. This expansion of computer users meant that developers could no longer make assumptions about their users, forcing them to consider design more carefully and begin thinking more seriously about usability testing.

Several studies in the computer engineering literature show the benefits of usability testing during the past decade. For example, Boeing officials recently reported that design modifications to their computer applications, resulting from usability testing, reduced their costs by approximately \$45 million (Thibodeau, 2002). Barnum (2002) reported that after making significant changes to the interface of computers at the New York Stock Exchange, based on usability testing, productivity doubled and error rates decreased by a factor of ten. Also, usability testing of the computer systems at American Express and the subsequent design changes derived from the testing resulted in a 90% decrease in the amount of time required to train personnel to use the systems, and a 75% decrease in the amount of time to complete tasks on the system

(Gibbs, 1997). After updating the design of the Dec Rally programming application based on usability testing, sales of the subsequent release were 80% higher than the version that was not usability tested (Wixon & Jones, 1992), and users reported usability as one of the primary reasons for purchasing the product over others.

Usability Testing Methods and Measures

Even the most basic usability test requires some degree of planning, preparation, and resource allocation. Those within the educational community (e.g., teachers, principals, administrators, and technology coordinators) lack the time, resources, and expertise to conduct rigorous usability tests on potential technology products before making a purchase. Therefore, a basic understanding of usability testing procedures and measures is necessary when assessing a product from a vendor. This section provides a general overview of common usability testing methods and the types of data collected.

Methods. In the current industry, usability testing is ubiquitous among software developers. However, there is little consistency between developers regarding the measures used during testing, when the testing takes place, and how to use the data. For example, Company A may conduct usability testing by surveying current users about their satisfaction with the product, while Company B may directly observe users interacting with the product and measure the time it takes to complete tasks, number of errors made, references to the user manual, etc. Even if developers use the same usability measures, they may not use them for the same purpose. For example, conducting usability testing after the product has been fully developed tends to serve more as 'Quality Assurance' and feedback for the next version, whereas usability testing conducted throughout the design process helps guide designers through an iterative process of development that is more likely to result in a product with optimal usability. The following

provides a brief overview of the methods and measures used in usability testing. More comprehensive descriptions of procedures can be found in Barnum (2002), Rubin (1994), and Dumas and Redish (1999).

Methods of usability testing vary widely depending primarily on the amount of time and money available to the developer. Chang and Dillon (1997) estimated that traditional usability testing can range from \$20,000 to over \$100,000, depending on how the tests are conducted and the system(s) under evaluation. This type of usability testing typically involves large numbers of participants observed in sophisticated labs with complex observation and recording equipment, followed by statistical analyses of the data (Barnum, 2002). However, the past decade has seen a shift from large-scale usability ‘experiments’ to smaller investigations with carefully chosen participants who are observed under more naturalistic conditions. Nielsen (1994), a pioneer of this ‘discount usability’ movement, asserted that the costs of running tests with more than five participants exceed the benefits gained. Acceptance of this type of usability testing by the HCI community made usability testing more feasible for smaller companies and freelance developers. However, small-scale usability tests also increase the importance of participant selection because the sample must reflect a range of demographics within the target population (e.g., novice, intermediate, and advanced computer users) (Brink, Gergle, Wood, 2001). Finally, users’ subjective ratings and impressions of the application are critical to a complete usability test. These are usually collected immediately following the usability test with Likert-scale ratings and one-on-one interviews, in addition to follow-up interviews/surveys that assess what features users remember and to what degree they remember how to use them.

One of the most frequently cited reasons for developers to refrain from usability testing is its cost -- not only the monetary costs, but also the time and effort. Large corporations such as

Apple and Microsoft have the resources to invest in this type of endeavor; however, smaller companies and/or individuals usually do not, even if it may lead to long-term savings. For this reason, technology has begun to play a larger role in usability testing. For example, Chang and Dillon (1997) developed the Automated Usability Software (AUS) to collect usability data without human observers. The software is installed on the user's computer and operates in the system's "background," tracking and timing every interaction the user has with the computer (e.g., mouse clicks and movements, keystrokes, etc.) The program compiles the data into various formats, such as graphs, scatter plots, and data sheets. The primary disadvantage of this type of testing is the lack of knowledge about the environmental contexts under which users experienced the system (e.g., physical location, distractions present, availability of external help from a manual or person, user affect, etc.). If the application is web-based, some degree of usability data can be gathered through server log files, but these data usually only describe user navigation patterns, time spent in specific areas of the website, and the users' system specifications.

Even if a developer failed to conduct any formal usability tests, 'mature' products that have been on the market for a year or more have experienced a number of 'informal' usability tests with actual users. Tracking the content of technical support communications helps develop an inventory of user problems and questions. Logs from technical support communications can provide detailed accounts of problems users experience under natural conditions and the type of help required to solve the problem. The primary limitation of using technical support data exclusively is the restricted sample of users who communicate with technical support centers. Some users, particularly novice users, avoid technical support. This can happen for a variety of reasons, including a lack of time, being unaware of the availability of these services, not

knowing how to make contact with these services, and/or being intimidated by technical support staff.

Measures. Although there is a wide range of usability testing methods, the range of variables measured in such testing is more constrained. In a typical usability test, the tester wants to know how long it takes users to complete tasks, how many errors users make, how much help they need, users' subjective ratings of the system, and, if the application requires training, how much training was needed. To answer these questions, the tester conducts detailed observations of users completing tasks, recording all user interactions with the system (e.g., mouse clicks and keystrokes), when they occurred, what the user said during the test, requests for help (e.g., references to a help manual, verbal requests, etc.), facial expressions, and responses to post-surveys/interviews. These measures can be grouped into three categories: Errors, time, and subjective responses.

Errors are perhaps the simplest form of usability data to analyze because fewer errors almost certainly lead to improved usability. However, simply reporting the 'raw' number of errors per task provides limited information. To provide context, errors are often reported as a ratio to correct interactions and/or a percentage of total interactions. As much as we would like to use software that elicits *no errors*, this type of software simply does not exist. Therefore, it is important to know how users responded when they made errors. Did they know they made an error? Were they able to self-correct? How long did it take to correct? How 'severe' was the error (e.g., did it prevent further progress, delay progress, etc.)? Were they able to correct the error using the Help menu or manual, or did they have to ask for assistance? In addition to assessing the usability of an application, error data provides information that can help technology coordinators prepare training and plan for additional in-house support that might incur as a result

of the application's implementation. For instance, if several problems are related to conflicts with the operating system or server used by the purchaser's school or district, discussion of these problems (and their solutions) should be integrated into training.

Time is usually used to determine how long it takes users to complete tasks, find information, or answer questions related to the application under evaluation. Because technology is often perceived as a way to improve efficiency and productivity, companies often present enticing claims that profess the time-savings that will result from using their application. Upon further evaluation, these claims usually amount to little more than broad estimates or anecdotes from advanced users. Time data from usability tests should provide specific time measures presented across a variety of tasks/subtasks for several participants with clear descriptions of the context of the usability test. In many cases, it is also helpful to compare the time it takes to complete tasks across several similar applications. For example, little is gained from knowing that teachers who used Application X could generate a complete roster of students in five minutes. However, knowing that a comparison group of teachers took 10 minutes to generate a roster with the same information using Application Y provides a point of reference.

Subjective responses and attitudes towards an application are measured with surveys containing Likert-type rating scales, interviews with users, and observations of users' facial expressions and verbal comments during usability testing. The most commonly reported usability metric, 'user attitudes', is often viewed as the most important measure of usability. However, usability testing can often reveal conflicts between users' attitudes towards the system and the more objective usability measures described above (Lagier, 2002). For example, in an evaluation of the usability of an online foster parent training system (Buzhardt, Heitzman-Powell, 2004), we found that the system received an average overall usability rating of 4.4 (five being the highest),

despite the inability of some users to complete critical tasks without assistance. Several factors over which the tester has little control can influence user ratings, including variations in users' definitions of usability, and some users' desire to please the tester (particularly if they are paid for participation or if they have a personal relationship with the tester). Despite these shortcomings, 'user attitudes' remains the cornerstone of a complete usability test because it provides insight into how users feel about the application, which ultimately affects their acceptance, continued use, and patience in working through potential problems.

Finally, the most critical (and the most frequently omitted) piece to assessing usability data, is the context in which the data were collected and the type of users who participated in the usability tests. For example, an application intended for use by teachers during day-to-day classroom activities should be tested with teachers in a classroom rather than college undergraduates in a controlled testing lab. Although the latter will, nonetheless, provide valuable information, the lack of an authentic testing environment and participants from the target population leads one to question the validity of the data. The context of a usability test also includes the training and preparation that participants received prior to the usability test. Any preparation or training that participants received should be clearly stated in any usability testing report, including participants' prior use of the application and/or involvement in its development through participation in focus groups or surveys.

The Educational Community's Role in Usability Testing

Although the software industry has seen a steady rise in usability testing during the past decade, consumers are often either unaware of the results or only receive the 'good news' from the evaluations (e.g., "9 out of 10 teachers gave it a five-star rating"). When a developer does conduct usability tests, unlike traditional research, the results remain internal and are rarely

shared with those outside the organization, nor do they undergo an independent peer review. Keeping usability findings internal often makes sense in order to improve usability, while maintaining costs. However, consumers' ability to make informed technology purchases would be significantly enhanced if developers made the findings of usability tests, or lack thereof, easily available. Fortunately, there are ways to infer the findings of usability tests (formal or informal) in the absence of formal reports. In this section, we discuss ways of judging a product's usability through secondary sources when the developer fails to provide a usability report, and how the educational community can encourage developers to conduct usability testing and make findings easily accessible.

Secondary Sources of Findings from Usability Tests

Developers rarely report errors experienced by users during usability testing for fear that it emphasizes the product's shortcomings. At the same time, they understand that providing consumers with information about known problems and solutions can lessen technical support costs. If the product has undergone any degree of testing with users, (whether anecdotal or rigorous tests with large numbers of users) the developers' interpretation of these findings will often be revealed indirectly in the product's documentation (e.g., user manuals, web sites, and/or help menus). These resources obviously help users install any necessary software, learn the system and troubleshoot problems that arise. However, for the organization that develops and sells the technology, these resources can directly affect the size and cost of their technical support network (Vilas, 2003). Therefore, the developer has a vested interest in producing the most effective product documentation in terms of helping users learn how to use the system and solve problems on their own.

Using product documentation as a window into the product's usability is often vague, but may be as much as the developer is willing to divulge. For example, known problems that emerge from usability tests and/or user feedback through technical support communications are often disguised as "Troubleshooting Tips" at the back of the user manual or Frequently Asked Questions (FAQs). FAQs frequently address general questions about the purpose of the application, who should use it, and how to get a demo version. The inclusion of specific questions about how to use the software (e.g., "How do I export data to a spreadsheet?", "How do I print weekly progress reports?") is often informed by some form of usability testing even if it was just anecdotal observations. The troubleshooting section of a user manual provides a clearer picture of what the developers have learned from usability tests, technical support history, and/or information about known "bugs" in the system. For example, a quick perusal of the Microsoft Windows troubleshooting documentation shows problems by category, many of which have likely been chosen based on results of usability tests. For example, if usability tests showed that 4 out of 5 users could not find an electronic roster they had just created, the troubleshooting guide may include "Lost Electronic Rosters" and steps for locating rosters.

Unfortunately, we often discover that user manuals and "Online Help Desks" are as difficult to use as the applications themselves. Many of the major software developers, such as Microsoft, conduct usability tests specifically on these user-assistance resources (Simpson, 1990), but many do not. Thus, in addition to looking for information about specific usability problems, consumers should assess support documentation to determine how well it will help them correct their own problems and answer questions without additional assistance. Some basic issues to consider include the following: Is the documentation is easily accessible in a variety of formats (e.g., book, online, electronically within the program, etc.) to accommodate user

preferences? Are the solutions stated in clear, non-technical terms? Is the documentation specific to the current version of the software or does it frequently refer to earlier or generic versions? When assessing these resources, keep a simple rule in mind: If you cannot understand it, it is unlikely that your teachers or students will understand it.

Encouraging More Usability Testing and Reporting

Other industries have forced the issue of usability testing by making usability data a primary consideration in their technology purchases, and perhaps most importantly, making this known to technology developers and distributors. For example, after finding that improving the usability of a productivity application saved the company nearly \$45 million, Boeing now requires developers to provide usability testing results before making a purchase. Furthermore, the National Institute of Standards and Technology developed a standard format for reporting usability data called the *Common Industry Format for Usability Test Reports* (see website, <http://zing.ncsl.nist.gov/iust/> for more information). The format requires, for example, objective usability data (e.g., time, error rates, requests for help, etc.) in addition to subjective ratings, and that the usability testing methods be detailed enough to replicate (Thibodeau, 2002). Although the availability of this standardized reporting format represents a positive move in terms of improving usability testing and dissemination, developers are not required to be aware of this standard, much less to follow it. It is ultimately the responsibility of consumers to encourage the use of the standard, or at least something like it. Microsoft, which already provides extensive access to their usability testing (see website www.microsoft.com/usability/), has expressed interest in using the standard, but says that *adopting this standard depends on whether or not consumers request it*.

Before we will see improvements in educational technology usability, the educational community must put more pressure on developers to conduct and report the results of usability tests. To be most effective, this pressure should come in the form of making technology purchases based in part on usability testing reports, and, whenever possible, making this known to developers. Although major developers, such as Microsoft and Apple, provide extensive access to information about their usability tests, smaller developers may not, but this does not mean the data do not exist. Sales representatives rarely know the details of usability tests, and may make references to user testimonies and surveys when asked about usability. If sales representatives do not have immediate access to these data or reports, they should be able to provide indirect sources for this information. For districts and schools that have existing contracts with technology providers, renewed contracts should include provisions that future purchases will be based partly on results from usability tests.

One could argue that integrating technology into schools will have little or no effect on students' learning outcomes. Nonetheless, given the technology initiatives of No Child Left Behind (US Dept of Education, 2002), the amount of professional development time devoted to technology training, and the millions of dollars that districts spend annually on technology, technology in schools is here to stay. The question becomes this: Will schools take full advantage of technology's potential, or will technology continue to suck valuable time and resources from educators who use applications that are needlessly difficult to learn and lack the usability to operate in the context of a busy classroom. Improving the usability of educational technology will not miraculously solve all of the difficulties associated with integrating technology into schools, but it is a piece of the puzzle that rarely receives attention by the educational community. Understanding usability testing methods and measures, looking for

evidence of these measures in product documentation before buying, and making usability test reports part of the purchasing decision process will, in the short-term, help ensure that schools use technology with known usability standards; and in the long-term it may increase the implementation of usability testing and reporting by technology developers.

Contributors

Jay Buzhardt, PhD, is an Assistant Research Professor from the Juniper Gardens Children's Project at the University of Kansas and Vice-President of Instructional Technology at Integrated Behavioral Technologies, Inc. His research program focuses on the development and evaluation of instructional and assessment technology for implementation in natural settings. Current projects involve the development of an online foster parent training system, online training for individuals providing behavioral treatment to children with Autism, and an online progress monitoring system to track the development of children 0-3.

Linda S. Heitzman-Powell, Ph.D., is an Assistant Research Professor at the University of Kansas, Juniper Gardens Children's Project, Kansas City, KS and President/Founder of Integrated Behavioral Technologies, Inc. Eudora, KS.

References

- Barnum, C. (2002). *Usability Testing and Research*. New York: Allyn and Bacon.
- Bottino, R., Forcheri, P., & Molino, M. (1998). Technology transfer in schools: From research to innovation. *British Journal of Educational Technology*, 29(2), 163-172.
- Brink, T., Bergle, D., Wood, S. (2001). *Usability for the Web: Designing Highly Usable Websites*. San Francisco: Morgan Kaufmann.
- Buzhardt, J. & Heitzman-Powell, L. (2004, May). *Development and Assessment of an Online Training System for Foster Parents*. Presentation at the 31st Annual Association for Behavior Analysis, Boston, MA.
- Change, E. and Dillon, T. S. Automated usability testing. In Proceedings of *Interact '97*.
- Coley, R., Cradler, J., & Engel, P. (1998). *Computers and classrooms: The status of technology in U.S. schools*. Princeton, NJ: Educational Testing Service.

- Cuban, L. (2001). *Oversold and Underused: Computers in the Classroom*. Cambridge, MA: Harvard University Press.
- Dumas, J., and Redish, J. (1999). *A Practical Guide to Usability Testing*. United Kingdom: Intellect, Ltd.
- Gibbs, W. (July, 1997). Taking computers to task. *Scientific American*, 82-89.
- Ike, C. (1997). Development through educational technology: implications for teacher personality and peer collaboration. *Journal of Instructional Psychology*, 24, 42-49.
- Lagier, J. (2002). *Measuring Usage and Usability of Online Databases at Hartnell College: An Evaluation of Selected Electronic Resources*. Unpublished Doctoral Dissertation from Nova Southeastern University.
- Machrone, B. (2004). MP3 Players: Apple iPod. *PC Magazine*. Retrieved May 5, 2005 from <http://www.pcmag.com/article2/0,1759,1634140,00.asp>
- National Center for Education Statistics (2005). *Internet Access in U.S. Public Schools and Classrooms: 1994–2003*. Washington, DC: Department of Education.
- Nielsen, J. (1993). *Usability Engineering*. Boston: Academic Press.
- Nielsen, J. (1994). Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. In R. Bias & D. Mayhew (Eds.), *Cost-Justifying Usability*, 242-272. Boston: Academic Press.
- Office of the Press Secretary (2000). *The Clinton-Gore Administration: A National Call to Action to Close the Digital Divide*. Washington, DC: The White House. Retrieved May 1, 2005 from <http://clinton4.nara.gov/WH/New/html/20000404.html>
- Park, J. and Staesina, L. (2004). Tracking US trends. *Education Week*, 23(35) 64-67.

- Rubin, J. (1994). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. New York: John Wiley & Sons, Inc.
- Simpson, M. (1990). How usability testing can aid the development of online documentation. *Proceedings of the 8th Annual International Conference on Systems Documentation*, 41-48.
- Thibodeau, P. (2002). Users Begin to Demand Software Usability Tests. *ComputerWorld*. Retrieved May 5, 2005 from <http://www.computerworld.com/softwaretopics/software/story/0,10801,76154,00.html>
- U.S. Department of Education (2002). *Questions and Answers on No Child Left Behind*. Retrieved May 3, 2005 from <http://www.ed.gov/nclb/methods/whatworks/doing.html>
- Stone, C. (1998). Overcoming resistance to technology. *The Delta Kappa Gamma Bulletin*, 64, 2, 15-19.
- Swain, C., Pearson, T. (2002). Educators and Technology Standards: Influencing the Digital Divide. *Journal of Research on Technology in Education*, 34, 3, 326-335.
- Vilas, A., Gonzalez, J.A., Gonzalez, B., Gonzalez, J. (2003). Digital learning-teaching environments and contents. *Journal of Digital Contents*, 2, 1, 1-10.
- Wixon, D. & Jones, S. (1992). *Usability for fun and profit: A case study of the design of DEC RALLY, version 2*. Digital Equipment Corporation.

Acknowledgement: The authors would like to thank Tim Johnson for his insightful comments on an earlier draft of this manuscript.