Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2003 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2003

A Decision Support System to Determine the Need and Extent of Training: A Case Study of Computer Literacy Training at a University

Paul Stephens
Bradley University

Amanda Youck
Bradley University

Follow this and additional works at: http://aisel.aisnet.org/amcis2003

Recommended Citation

Stephens, Paul and Youck, Amanda, "A Decision Support System to Determine the Need and Extent of Training: A Case Study of Computer Literacy Training at a University" (2003). *AMCIS 2003 Proceedings*. 410.
http://aisel.aisnet.org/amcis2003/410

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2003 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A DECISION SUPPORT SYSTEM TO DETERMINE THE NEED AND EXTENT OF TRAINING: A CASE STUDY OF COMPUTER LITERACY TRAINING AT A UNIVERSITY

Paul Stephens
Foster College of Business Administration
Bradley University
prs@bradley.edu

Amanda Youck
Foster College of Business Administration
Bradley University
ayouck@bradley.edu

Abstract

The ubiquity of computer technology requires that individuals adopt and use technology. There are individuals in every organization that do not possess the skills necessary to adopt and use technology. Therefore, training is necessary for individuals to take advantage of technology. Organizations need to know which individuals need training. There is a significant and positive relationship between computer self-efficacy and the adoption and use of computer technology. Therefore, many individuals with low computer self-efficacy need training that will improve their self-efficacy and provide them with the skills to adopt and use technology. Self-selection to train does not work because individuals with low computer self-efficacy will avoid training. A decision support system can be used to quickly identify individuals with low computer self-efficacy so that they can be routed into training. The system not only determines which individuals require training but also determines which type of training is most appropriate. The DSS provides cost benefits by eliminating redundant resources, reducing the number of instructors, streamlining the administration of training programs and identifying those individuals that do not require additional training. The concept of a decision support system built upon self-efficacy can be applied in a variety of environments beyond computer literacy training.

Keywords: Decision support systems, training, computer self-efficacy, end-user computing

Introduction

The ubiquitous nature of computer technology throughout organizations requires that individuals adopt and use technology in order to be successful in their endeavors. The information age is upon us and computers, the enablers of information literacy, are significant parts of our business life and culture. A dialogue amongst researchers and educators has been ongoing for years about how to prepare people for the information age via computer literacy (Haigh 1985, Jungck 1990, Jones and Pearson 1996, Breuch 2002). Today, researchers and academics are no longer the sole source advocating information literacy. The discussion of the digital divide is widespread. Business leaders, politicians and the mainstream press are arguing that in order for people to live productive lives, we all need to be comfortable with and knowledgeable about technology (Palm Beach Post 2002, National Journal Group 2001, Koretz 2002).

Given the call for information literacy, it would seem that computer and information literacy training is integrated throughout our educational systems. Unfortunately, there exits many areas throughout the world, including the United States, where computer and information literacy education is simply non-existent (Swain and Pearson 2003, Tipton 2002, Moeck 2002). The digital divide is real and there are individuals in every organization that do not possess the skills necessary to adopt and use technology. Therefore, computer vocational training is needed in every organization. The main goal in vocational training is to teach new skills or to upgrade existing skills. The need for training creates a need to manage the costs associated with training. Costs associated with vocational training include administration, trainee time, instructor time, labs, equipment, and instructional materials. Many of these resources have to be continually updated to ever changing technological platforms (Tsang 1997). It would be easy to simply force everyone through training programs. Costs associated with deciding whom to train would become nominal. But, the majority of other training costs skyrocket because we needlessly provide training to people who don't need it. This is a waste of resources and breeds resentment amongst employees that do not require training. So, most organizations must find a way to determine which people need training while attempting to minimize administrative expenses. The easiest option is to allow individuals to self-select. Organizations may choose this route but it has significant drawbacks: Individuals that need the training will select out, individuals that don't believe they need training but they actually do will select out and individuals that don't need training will choose to go through the training anyway. These choices are made for a variety of reasons. How people react to computing technology has been well researched (Davis 1989, Taylor & Todd 1995, Compeau et al 1999). Compeau et al (1999) uses social cognitive theory to explain that one cognitive factor (self-efficacy) has a significant impact on individual behavior towards technology. Computer self-efficacy (CSE) can be defined as an individuals' belief in their ability to use technology in order to solve problems, make decisions, and to gather and disseminate information (Stephens and Shotick 2002). Research has found that computer self-efficacy (1) is positively and significantly related to the early adoption of technology (Burkhart and Brass 1990), (2) is significantly related to students registering for computer courses at universities (Hill 1987), (3) has an impact on performance in learning and applying technology (Compeau et al 1995) and (4) allows individuals to envision how technology can be used to improve their work related performance (Stephens and Shotick 2002). When confronted with choice, individuals with low computer self-efficacy will avoid computer technology training. Therefore, mandatory training is required for individuals that need to be introduced to these skills.

Since self-selection is not an effective option, organizations are still confronted with the problems of who should be trained and which method of training is best suited to any particular individual. In an academic environment, the problem of who to train is made even more difficult. If you make a class optional for students, most will simply opt out. The university that we chose for our case study has a policy that every student entering the college of business should possess basic computer literacy. Research at this institution has shown that they cannot assume that every student entering the college has computer literacy skills. In research conducted in 2001, it was found that the computer literacy skills of 25% of an incoming freshman class were significantly deficient while another 25% could be categorized as marginally literate. On the other hand, 25% of the incoming students had extensive training in computer literacy (Stephens & Shotick 2001). Of course, there is the possibility that those who had no formal training have picked up significant skills on their own and it is also possible that those who report significant training have not retained the skills they were taught. So, the college policy is for the student to demonstrate computer literacy in those skills needed throughout their coursework. There are three distinct paths available for the student: (1) pass a series of three, hands-on exams that demonstrate computer literacy, (2) complete an on-line course in computer literacy skills or (3) complete an in-lab class in computer literacy skills. The college has implemented a student choice policy where students select which path they would prefer to take. This policy has created a situation that requires many resources. To support the program, the college offers six to seven sections of a one-hour class each semester. Additionally, a database is used to track students who don't take the class and students who don't complete the requirement in their first year are encumbered for registration until they take the exams or sign up for the class. Several exam slots are made available each semester for students to take the proficiency tests. It takes three people to directly manage the system and several support staff are involved in peripheral issues. The college has been searching for a solution that will serve the needs of their students while minimizing the resources needed to implement the program. Additionally, internal research has found that the student choice policy just doesn't work as expected. Since a significant proportion of the incoming students report high CSE, it would make sense that these students would choose to take the proficiency exams and avoid the work involved with taking a class. But, the college has found the opposite to be true. Some 90% of the incoming students end up taking the on-line or in-lab class. Many students avoid the requirement until they are forced into the class. So, the college finds itself back at square one.

Many organizations, including the college, need a tool that will allow them to identify who needs training and what kind of training is most appropriate. How does an organization determine who to train and how does the organization avoid the problems of self-selection? The purpose of this paper is to discuss a decision support system that has been developed for the college of business that answers both those questions. The paper will show how and why this current system has been developed. Finally, the paper will show that this methodology has implications beyond an academic environment.

Computer Literacy Training at the University

There was a time when it was necessary to provide computer literacy training for all university students. As primary and secondary school systems began to offer computer skills training, computer literacy training in post secondary institutions (colleges and universities) started to become redundant. A decade ago, expectations were that within a few years, the vast majority of incoming students would arrive at college with basic computer literacy. Instead, research continued to find that students were not coming to the university with basic computer literacy (Sheffield 1998). As we discussed earlier, our research indicates that the need for computer literacy training of incoming university students is still necessary for a significant portion of those students. There are many reasons why students are coming to the university with inadequate computer literacy skills.

Initial interviews with five high school curriculum designers reveal two areas of concern. Literacy skills are not a requirement for college bound students. First, if a student chooses to avoid technology they can. In fact, students who are concerned about their grade point average are often discouraged from taking computer technology classes. These classes are not normally associated with the college prep program and earning an A in the class is worth fewer grade points than earning an A in the college prep program. Secondly, literacy skills are not being uniformly reinforced in other classes. Many teachers are not incorporating technology into their curriculum. This stems mainly from a lack of knowledge on the part of the teachers themselves. Training is available but teachers are slow to get training and then even slower to incorporate lessons into the classroom. There is some resistance from teachers but mainly it is a lack of time that is hampering the process. Additionally, there is a lack of technological resources throughout the schools. Many schools have isolated pockets of technology available for use but it is difficult to incorporate them into every day use in the classroom or even for routine assignments.

The interviews we have conducted up to this point could be considered "good case" scenarios. The school districts are fairly well funded and the communities fairly affluent. One would expect to find some drastic differences from community to community with low-income school districts showing a complete lack of literacy training and high-income school districts showing an abundance of resources.

Deciding Who to Train

It would seem to be a simple solution to test all students prior to the start of classes to determine if they are exempt from the class or need to take the class. This solution presents two problems: (1) the testing is time consuming and expensive and more importantly (2) for those students that lack computer skills, failure has a prolonged negative effect upon their computer self-efficacy. A person's introductory interaction with computers is a source of self-efficacy information. (Sein, Bostrom, and Olfman (1987)) Creating an environment in which failure is guaranteed is not desirable. Therefore, we need to use a method for determining computer literacy that does not inflict prolonged damage to the student's self-efficacy. Traditionally, computer self-efficacy scales have been developed and used to determine the effectiveness of literacy training. (Karsten & Roth, 1998) But, this paper argues that these scales can be used to decide if training is necessary and what kind of training is appropriate.

Computer Self-Efficacy (CSE)

Earlier in the paper, the concept of computer self-efficacy (CSE) was defined broadly. The concept has been expanded in the literature to include three distinct types of CSE: General Computer, Task Specific and Profession Oriented CSE (Marakas 1998, Stephens and Shotick 2002). Profession oriented computer self-efficacy (PCSE) refers to an individuals' judgment of efficacy across computer applications that comprise those needed to work in a particular profession. Computer literacy expectations for an artist should be far different than what one should expect for an accountant. Different professions and/or academic units would include different skills in their self-efficacy scales. A task included in engineering might entail the ability to use CAD (Computer Aided Design) software, in advertising a potential task is the ability to use publishing software to create brochures, in multi-media it might mean the ability to use simple photo editing software to incorporate images in electronic media, in chemistry it could include the ability to interact with instrumentation software in the lab. Therefore, each specialized professional area should create a specific scale targeting the computer related skills for that vocation. Researchers can then use these specialized scales to do targeted research in specific professions. Researchers must consider the type of CSE they are studying, the characteristics of the individuals in their sample and use the appropriate self-efficacy tool.

Computer Self-Efficacy Scales

Researchers have developed several CSE scales in order to examine the relationship between CSE and human behavior with respect to using computers. One article in particular provides us with a summary of the variety of scales that have been developed over the years (Marakas 1998). Instruments designed to measure CSE tend not to differentiate on the distinction of different types of CSE, sometimes incorporating multiple types of self-efficacy while ignoring others. Given our discussion of the different types of CSE, there should be one scale that measures general computer self-efficacy and multiple scales focusing on the different types of PCSE and TCSE.

The Business Computer Self-Efficacy Scale (BCSE)

For the purposes of this research, we have used a PCSE scale known as the business computer self efficacy scale (Stephens & Shotick 2002). The scale targets those skills that are important for careers emerging from the typical college of business. The complete scale can be found at **http://bobcat.bradley.edu/~prs/bcse.htm**. The BCSE Scale takes a specific skill set and applies it to a targeted audience. Following the guidelines established by researchers in the field of self-efficacy, the scale uses a composite measure of magnitude and strength (Bandura 1977). Magnitude scales, which are commonly used in social science research, attempt to determine the direction (magnitude) and strength of people's beliefs (Lodge 1981). This form of self-efficacy scale has been shown to provide the best correlations with goals and performance in research (Lee & Bobko 1994). For this research, we are using the scale for incoming freshman business majors.

A DSS Built Around Computer Self-Efficacy

To automate the delivery of the scale to incoming freshmen, we built an interactive decision support system (DSS). The DSS collects information from students and calculates a computer self-efficacy score. The back-end database specifies the route that students must take in order to demonstrate or learn those computer literacy skills required for the college of business. In this section, we discuss the tool itself and how it works. In the sections that follow, the paper discusses the application of the DSS, the benefits accrued and how similar systems could be implemented in other settings.

Upon evaluating the needs that the application, it was concluded that the use of Microsoft Access augmented with Visual Basic (VB) would allow us to create the tool we needed. This is due to the many features that Access and VB provides that support design and stability. It was determined that a product would need to be developed that is user friendly for both the administrator and the student. Subsequent usability testing allowed us to refine the program to maximize the utility of the application. Three groups of students (5 students per group) were brought into the lab and asked to run the application. We used a mix of students with both high and low computer self-efficacy. Student interactions were observed (two observers were used during testing) while using the application. Additionally, students provided written feedback about the application. Subsequently, changes were made in application that impacted both the simplicity of the application and the reliability of the data collected. For example, if the student stated that they could not perform a certain task, they are not supposed to assign a ranking of confidence in their answer. Some students ignored the instructions and picked a confidence rating in all cases. We changed the program to adjust all no answers to a ranking of zero for the purposes of calculating a score. Another error was that when the student picked the back button on the first question, the program failed. This problem was corrected. The application was locked down so that students could not break out of the application by mistake. We have been careful to design the tool so as not to introduce non-random error into the results. Our research has confirmed the reliability of the tool. Reliability was tested using the internal consistency method and the Cronbach alpha was found to be .901. Validity testing will be conducted as the tool is introduced for use and subsequent revisions will be made based on such testing. Scores from the tool will be compared to the performance of students on the proficiency exams and in the class to determine the predictive validity of the tool. The DSS includes interaction with an outside source file to store data. This provides the administrator with a wealth of data that can be analyzed on a yearly basis. The program is installed as a stand-alone product on individual computers in the testing lab. We have determined that this method will allow us to better control the process rather than making the application available via the web.

The interactive tool consists of a front-end application that collects student responses and a back-end database that stores all student responses and provides administrative tools. At the front end, the user will be asked a series of background questions consisting of name, gender, ID number, and related computer experience. Once this information is entered, the student will then proceed into the scale portion, where they are guided through several questions related to the assessment of their skills in completing various technological tasks. The front-end database has been assigned an autoexec macro that launches the function. It begins by requesting the students' identification number. If the student cancels the box, then an error will trigger the box to appear forcing the student to enter a number. If the student enters a number, the program assigns the number to the BUID field on the student table and to BUID field for all records on the answers table.

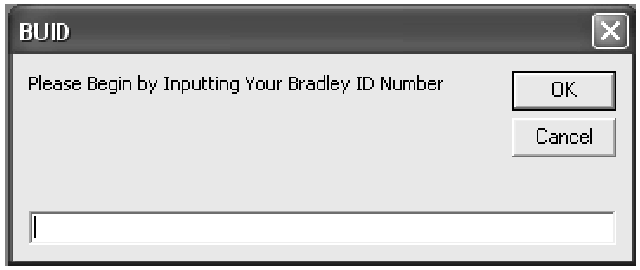

Figure 1. First Message Box Requiring the Student Identification Number

Thus, the relationship between the student table and answers table is automatically created. The program ends at the student form that the student will fill out and press continue.

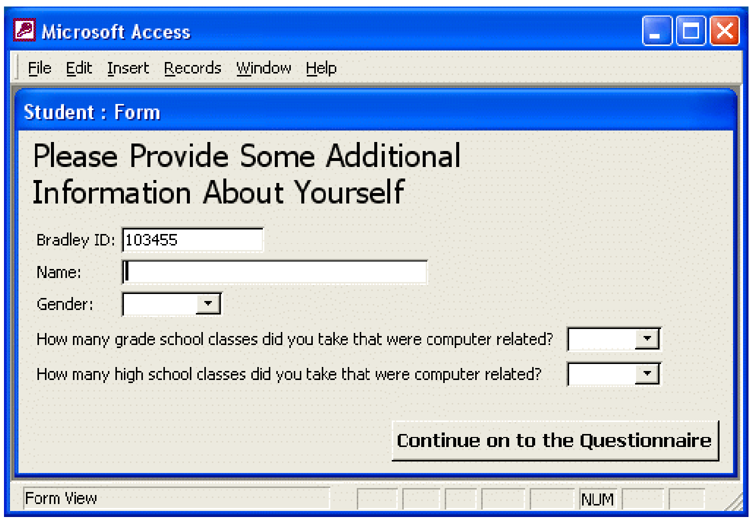

Figure 2. Student Information Form

The "Continue on .." button will hide the student form and show the questions form.

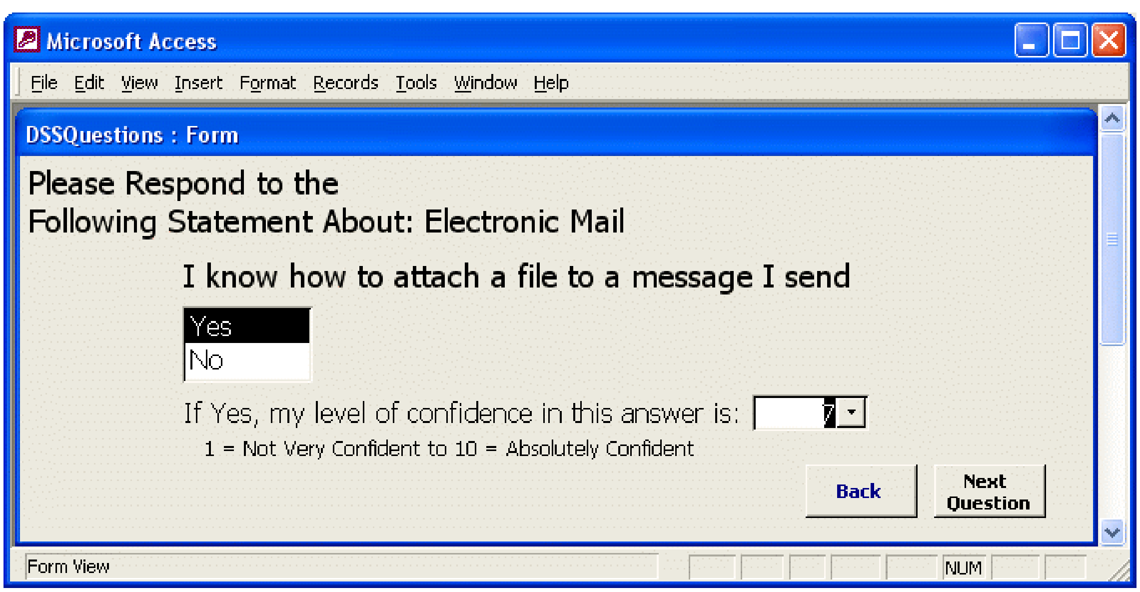

Figure 3. The Questions Form

The next question button will cycle the form through all of the questions. The back button allows the user to return to the previous question. When the form reaches the last question, it will close the form and open the Information form. The information form lets the student know they are done and when they will know the results of their questionnaire.

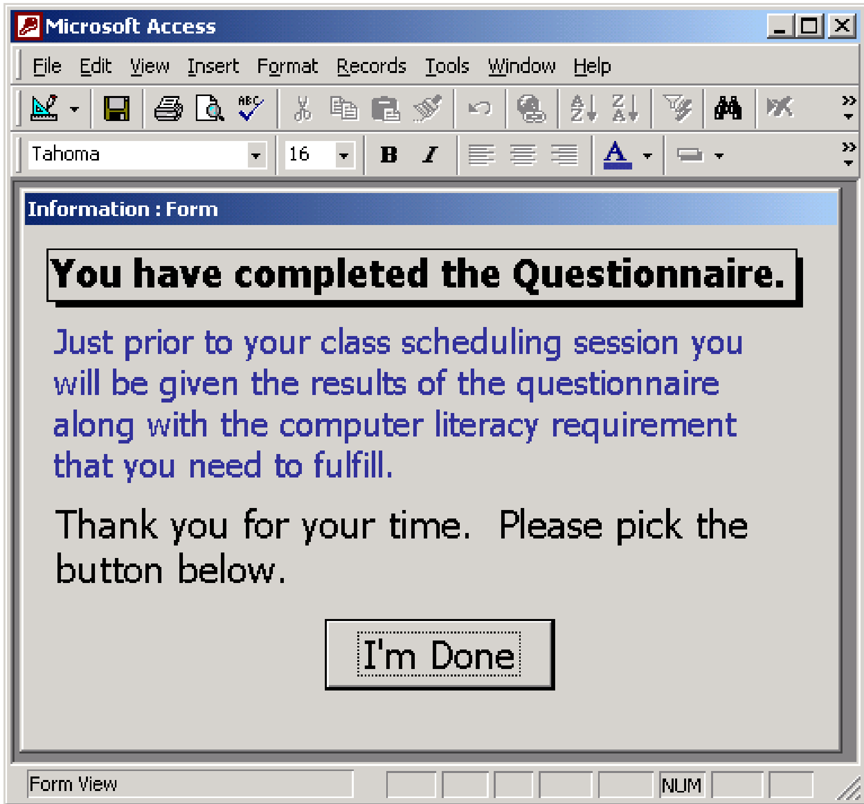

Figure 4. Informational Form

When the student picks the "I'm Done" button, the data for the student is exported out to the back-end database on the server, the tables are cleared and the application is closed.

The responses to the questions and the background data are stored in the backend database that is available for the administrator and can be used in a statistical analysis of the performance of the system. This will be discussed further in the section on the benefits of the DSS. The back end database compiles the results of the questions and outputs an assessment which details for the student which route towards computer literacy they must follow: (1) take a series of proficiency exams that prove literacy, (2) enroll in an on-line course that provides students the tools to complete a self-paced program in order to complete the proficiency exams that are part of the course or (3) enroll in an in-lab course where students will be provided an intimate learning environment where they will be taught the skills needed for computer literacy in the college.

Application of the DSS

To implement the system, it is required that we find a time and place where incoming business freshmen are brought together prior to the start of classes. For the university in this case study, the majority of students go through an orientation session during the summer prior to their first fall semester. It is during these orientation sessions that we have the students use the DSS. Orientation is structured so that students sign up for fall classes about halfway through the three-day session. The DSS is used early in the session, takes about 30 minutes and is held in a computer lab located in the college of business building. The results are delivered to the students during the time they are signing up for classes. The assessment tells the students whether they have to take proficiency exams or a class. If the students are required to take the exams, the assessment gives them the schedule for the time and place for the exams. We have targeted Friday afternoons starting about halfway through the first semester for the exams. This allows the students time to prepare for the exams. Preparation materials are made available on-line. Additionally, the Friday afternoon time slot doesn't interfere with classes because very few classes are offered during that time. For situations where a conflict with classes occurs, the administrator has to make arrangements with the student. Fortunately, these situations are extremely rare. For students that are required to take a class, the student will sign up for the class for the fall or spring semester of their freshman year.

Scoring the Responses

Scales based on Bandura's (1977) theoretical constructs capture both the strength and magnitude of the responses and provide general guidelines for assessing self-efficacy. Research has established a method for scoring such scales (Mcauley et al, 1991, Taylor et al, 1984). The scale asks the respondent whether they believe they can perform a specific task (magnitude). If the answer is no, then the response is scored as zero. If the answer is yes, then the participant is asked how certain they are in their yes answer on a Likert scale (strength). The strength scale might be 1-5, 1-7 or 1-10 and is normally spanned from highly uncertain to completely certain. The strength number from the Likert scale is used as the score for that response. Then the scores for all responses are simply added together and divided by the total number of items on the scale to attain a self-efficacy score. The flexibility of this method lies in capturing both magnitude and strength. For example, a participant who reports broad based magnitude (all yes responses) but moderate strength (5 out of 10) may score lower (a final score of 5 given 100 items) than a participant who reports sporadic magnitude (60% yes responses) with high certainty in those skills that they feel they can perform (10 out of 10 and a final score of 6).

Establishing the Thresholds

Gist et al. (1989) use computer self-efficacy as an estimate for general computer learning capability. They simply divide the scores into thirds: high, middle and low. Unlike our study, they were not using the divisions to determine different types of training but to ensure that treatment groups were not skewed. The authors were able to show that different training approaches are more effective for individuals with low *a priori* self-efficacy scores. Individuals with low computer self-efficacy require more instructor mediation and feedback on performance because anxiety and lack of confidence interact to prevent students from adopting a self-help approach (McMahon et al. 1999). Therefore, students that report relatively low computer self-efficacy should be required to take an in-lab class that provides this type of support. Clearly, students reporting high levels of computer self-efficacy require no further training and can simply take a series of proficiency exams that demonstrate their abilities. Those students that fall in between should take the self-paced, on-line course. The problem at hand was establishing the score thresholds for each category.

Incoming freshmen were administered the self-efficacy scale during orientation and their scores were determined. Students were explicitly told that they were required to take the computer literacy class or proficiency exams during their freshman year. The rule is documented in course catalogs and advisors are also asked to encourage students to complete the requirement. At the end the year, we tracked the students to discover if they chose to take the computer literacy class, took proficiency exams to substitute for the class or avoided doing either. We did not directly enforce the rule allowing students to decide for themselves what actions to take. The self-selection method presented some interesting results. As might be expected, a significant percentage of the students with low self-efficacy scores avoided the requirement but we also found that a significant percentage of students with high self-efficacy scores also avoided the requirement. Subsequent interviews with the high self-efficacy students discovered that the prevailing attitude amongst the students was that they had nothing to prove by taking the exams, they already knew the subject matter and they perceived little benefit from taking the time to come in and take exams. If we wanted proof of their skills, we would have to force them to take the exams. As was discussed earlier in the paper, for the system to work efficiently while minimizing administration costs, students must ushered through and not allowed to choose. The DSS provides that mechanism.

The data does provide us some insight into establishing the initial thresholds for the decision support system. There was a significant performance difference in the lower quartile between the in-lab class and the on-line class. (P=.01) Therefore, scores of 2 or less should be required to take the in-lab class. The third quartile performed the same in the on-line and in-lab classes and we decided that scores greater than 2 and up to 7 should be required to take the on-line course. Students in the second quartile had excellent performance regardless of the type of course they took. Although it was clear that some of these students should have taken the proficiency exams instead of taking a course. A significant portion of the students in the second quartile received almost perfect scores for the course. Therefore, we decided that for our initial thresholds, scores of 7 or greater should be required to take the proficiency exams, scores greater than 2 but less than 7 should be in the on-line class and scores of 2 or less should be in the in-lab class. At this point, we haven't used the DSS. It will be used for the first time in the summer of 2003. Our main concern is forcing students to take proficiency exams that are not prepared to do so. Therefore, we have set the thresholds at levels that we believe are conservative and that students who score 7 or better shouldn't have any problems passing the proficiency exams.

Benefits of the DSS

The main benefit of the DSS is that it allows the organization to determine which individuals need training and what type of training is required. The impact of this benefit is that it allows the organization to minimize the resources needed for training the user. Fewer instructors are needed because fewer class sections are required. For the college discussed in our case, class sections will be reduced from seven to two. As a result, two of the three people needed to manage the system should be able to be eliminated. The remaining person will be able to handle all the work necessary for the new system because some of the current administration and tracking tasks can be eliminated. Less staff time will be required to support the system as well. Significant cost savings are expected.

Another important benefit of the DSS is that it captures the data needed to determine if the system is operating as expected. On a yearly basis, scores can be compared to performance to determine if the tool remains both reliable and valid. Thresholds can also be adjusted via the analysis of the data.

The ability to capture data on a regular basis also allows for further analysis the computer self-efficacy data. This information can be used to assess differences in gender, age, and other mitigating factors. Additionally it provides a time series frame in which to examine the performance of individuals at different stages of their development.

For the college, at some point in time the DSS will be no longer needed. Incoming college students will eventually come to the university with the standard set of computer literacy skills. Therefore the use of the DSS will determine when basic computer literacy training is no longer necessary.

Adapting the DSS

The decision support system can be adapted to disciplines other than business computer applications. Various profession oriented CSE scales can be used to determine training needs in engineering, the sciences and other educational areas. Right now the tool is not extremely adaptable in this regard but could be improved to be flexible enough to offer different item sets and thresholds dependent upon the academic discipline. In fact, the DSS could use a general computer self-efficacy scale that could be applied to all incoming college students to determine if the institution still needs a basic computer literacy course required for graduation.

The decision support system developed for this case study has applications beyond deciding which students should take computer literacy courses. Various organizations could adapt the DSS to assess the training needs of their employees. Task specific self-efficacy scales can be developed for any training need. The DSS can then be used to determine which employees require training and the type of training that would be most beneficial. All of the cost advantages of using the DSS discussed previously are available to any firm that provides financial support for training or has in-house training programs.

Conclusions

The ubiquity of computer technology requires that individuals adopt and use technology. There are individuals in every organization that do not possess the skills necessary to adopt and use technology. Therefore, training is necessary for individuals to take advantage of technology. Organizations need to know which individuals need training.

There is a significant and positive relationship between computer self-efficacy and the adoption and use of computer technology. Therefore, many individuals with low computer self-efficacy need training that will improve their self-efficacy and provide them with the skills to adopt and use technology. Self-selection does not work because individuals with low computer self-efficacy will avoid training.

A decision support system can be used to quickly identify individuals with low computer self-efficacy so that they can be routed into mandatory training. The system not only determines which individuals require training but also determines which type of training is most appropriate.

The decision support system provides cost benefits by eliminating redundant resources, reducing the number of instructors, streamlining the administration of training programs and identifying those individuals that do not require additional training thus minimizing the expenses associated with training.

The concept of a decision support system built upon self-efficacy can be applied in a variety of environments beyond computer literacy training in the college of business. Further research in this area could determine the effectiveness and actual cost savings associated with using this type of tool in a variety of organizations.

References

Bandura, A. "Self-efficacy: Toward a Unifying Theory of Behavioral Change," Psychological Review (84:2), 1977, pp. 191 – 215.

Burkhardt, M. and Brass, D. "Changing Patterns or Patterns of Change: The Effects of a Change in Technology on Social Network Structure and Power," Administrative Science Quarterly (35), 1990, pp. 104 – 127.

Compeau, D. and Higgins, C. "Computer Self-Efficacy: Development of a Measure and Initial Test," MIS Quarterly (19:2), 1995, pp. 189 – 211.

Compeau, D., Higgins, C. and Huff, S. "Social Cognitive Theory and Individual Reactions to Computing Technology: A Longitudinal Study," MIS Quarterly (23:2), 1999, pp. 145 – 158.

Davis, F. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," MIS Quarterly (13:3), 1989, pp. 318 – 340.

Gist, M., Schwoerer, C. and Rosen, B. "Effects of Alternative Training Methods on Self-Efficacy and Performance in Computer Software," Journal of Applied Psychology (74:6), 1989, pp. 884 – 891.

Haigh, R.W. "Planning for Computer Literacy," Journal of Higher Education (56:2), 1985, pp. 161 – 171.

Hill, T., Smith, N., and Mann, M. "Role of Efficacy Expectations in Predicting the Decision to Use Advanced Technologies: The Case of Computers," Journal of Applied Psychology (72:2), 1987, pp. 392-414.

Jones, M. and Pearson, R. "Developing an instrument to measure computer literacy," Journal of Research on Computing in Education (29:1), 1996, pp. 17 – 28.

Jungck, S. "Viewing Computer Literacy Through a Critical, Ethnographic Lens," Theory Into Practice (29:4), 1990, pp. 283 – 289.

Karsten, R. and Roth, R. "The Relationship of Computer Experience and Computer Self-Efficacy to Performance in Introductory Computer Literacy Courses," Journal of Research on Computing in Education (31:1), 1998, pp. 14 – 24.

Breuch, L. "Thinking Critically about Technological Literacy: Developing a Framework to Guide Computer Pedagogy in Technical Communication," Technical Communication Quarterly (11:3), 2002, pp. 267 – 288.

Koretz, G. "Narrowing the Digital Divide", Business Week (3811), 2002, p. 28.

Lee, C. and Bobko, P. "Self-Efficacy Beliefs: Comparison of Five Measures," Journal of Applied Psychology (79:3), 1994, pp. 364 – 369.

Lodge, M. Magnitude Scaling: Quantitative Measurement of Opinions, London: Sage Publications, 1981.

McAuley, E., Wraith, S. and Duncan, T. "Self-Efficacy, Perceptions of Success, and Intrinsic Motivation for Exercise," Journal of Applied Social Psychology (21:2), 1991, pp. 139-155.

McMahon, J. Gardner, J., Gray, C. and Mulhern, G., "Barriers to Student Computer Usage: Staff and Student Perceptions", Journal of Computer Assisted Learning (15:4), 1999, pp. 302-311.

Marakas, G., Yi, M. and Johnson, R. "The Multilevel and Multifaceted Character of Computer Self-Efficacy: Toward Clarification of the Construct and an Integrative Framework for Research," Information Systems Research (9:2), 1998, pp. 126 – 163.

Moeck, P. "The Digital Divide and Rural Community Colleges: Problems and Prospects," Community College Journal of Research & Practice (26:3), 2002, pp. 207 – 224.

National Journal Group. "Democrats contrast high tech agenda with Bush approach," National Journal's Congress Daily, AM edition, 4/6/01.

Palm Beach Post, The. "Florida Grants $1 Million to Help Bridge State's Digital Divide," Dec 27, 2002.

Sein, M., Bostrom, R. and Olfman, L. "Training End-Users to Computers: Cognitive, Motivational, and Social Issues," Information Systems and Operations Research (25:3), 1987, pp. 236-254.

Sheffield, C. "A trend analysis of computer literacy skills of preservice teachers during six academic years," Journal of Technology and Teacher Education (6:2/3), 1998, pp. 105-114.

Stephens, P. and Shotick, J. "Computer Literacy and Incoming Business Students: Assessment, Design and Definition of a Skill Set," Issues in Information Systems (2), 2001, pp. 460 – 466.

Stephens, P. and Shotick, J. "Re-Evaluation of the Computer Self-Efficacy Model: Development and Use of the Business Computer Self-Efficacy Scale," Issues in Information Systems (3), 2002, pp. 251-257.
Swain, C. and Pearson, T. "Educators and Technology Standards: Influencing the Digital Divide," Journal of Research on Technology in Education (24:3), 2003, pp. 326 – 335.
Taylor, M.S., Locke, E., Lee C. "Type A Behavior and Faculty Research Productivity: What Are the Mechanisms?," Organizational Behavior & Human Performance (34:3), 1984, pp. 402-418.
Tayor, S. and Todd, P. "Assessing IT usage: The role of prior experience," MIS Quarterly (19:4), 1995, pp. 561 – 570.
Tipton, F. "Bridging the Digital Divide in Southeast Asia," ASEAN Economic Bulletin (19:1), 2002, pp. 83 – 99.
Tsang, M.C. "The Cost of Vocational Training." International Journal of Manpower (18:1/2), 1997, pp. 63-89.