Teaching User Interface Design: An Important Addition to the Information Systems Curriculum

Michael A. Eierman
University of Wisconsin Oshkosh, eierman@uwosh.edu

Jakob Iversen
University of Wisconsin Oshkosh, iversen@uwosh.edu

TEACHING USER INTERFACE DESIGN: AN IMPORTANT ADDITION TO THE INFORMATION SYSTEMS CURRICULUM

Michael A. Eierman[1]
College of Business Administration
University of Wisconsin Oshkosh
Oshkosh, WI 54901
(920) 424-0183
eierman@uwosh.edu

Jakob Iversen
College of Business Administration
University of Wisconsin Oshkosh
Oshkosh, WI 54901
(920) 424-0166
iversen@uwosh.edu

ABSTRACT

Information Systems programs have focused on educating future developers in analysis, design and programming. However, in most of these programs, design is overshadowed by the programming and analysis aspects of the curriculum. The interface is specified by the instructor to be implemented by the students, with little attention to how that interface should work for the users. The result is programmers that develop systems, which users find difficult to use to perform their jobs. Poor user interface design has a significant impact on user productivity and error rates, and an increased need for user support. In this paper, we (1) examine problems caused by poor interface design, (2) describe characteristics of good design, (3) describe a course on interface design, and (4) describe the results of teaching the proposed course.

KEYWORDS

Interface Design, GUI Design course, Information Systems Curriculum

INTRODUCTION

The United States software development industry is facing an unprecedented challenge from offshore development companies located primarily in Mexico and India. These companies offer lower cost (Carmel & Agarwal, 2001; Perez, 2003) and, in some instances, higher quality software products (Alanis, Mendoza & Alanis, 2004). Software quality can be measured in many ways. The two most commonly used are the absence of errors (bugs) and adherence to the program specifications (it correctly does its job). However, a third measure of software quality often is overlooked: how well the software works for the users. Even software that is of high quality in reliability and functional specifications can cost organizations a significant amount of money from lost productivity, increased error rates, increased training requirements and increased support needs if the system fails to account for the users and their context (Marcus, 2002). Software with a poor user interface design that is used by customers of the organization can lead to decreased sales, increased customer support costs, increased training costs, and a diminished perception of the company (Marcus, 2002; Raskin, 1997). Additionally, Seffah and Metzker

---

[1] Corresponding Author.

(2004) suggest that a well-designed interface can help users perform tasks more efficiently and directly impact organizational productivity and performance. Galitz (2002) summarizes a set of research on the benefits of good design, which finds that good interface design can reduce the time required by users to complete transactions by 25%, reduce errors by 25%, reduce decision-making time by 40%, and reduce the time to find required information by almost 50%. It is in the third area of software quality, interface design, where developers in the United States can develop a competitive advantage. In the long run, the U.S. economy may be better served if university programs in Information Systems increase their focus on software interface design.

THE COURSE

Focusing IS undergraduate programs on interface design does not mean forgetting traditional education requirements. Information systems professionals must know how to develop a robust, maintainable system that meets the user's needs. However, Galitz (2002) suggests that the information systems profession continues to produce such systems with interfaces that are confusing, inefficient, and sometimes just unusable. He suggests that the root cause is that IS professionals don't know what makes a good design. The undergraduate IS curriculum should address this problem. It currently does not. In most IS programming courses, the professor provides the student with an interface to create as a part of a programming assignment. The relative value of that interface is not questioned. The goal is to learn the mechanics of making the interface work as specified. If an IS curriculum addresses interface design, it is likely within the context of a systems analysis course where the topic receives limited attention among the myriad of other topics covered in the course. This treatment leaves the student with the impression that any interface will do. However, to raise the perception of interface design as an important aspect of system quality and to properly teach students how to develop quality interfaces requires a course that focuses wholly on design. The overall objective of the proposed course is to develop student understanding of the problems and benefits of good design, interface design objectives, principles and methods. The primary focus of the course is to develop this knowledge by letting the student make design decisions and then evaluating those decisions.

In developing the course, we examined the literature associated with design education and collected and examined course syllabi for design courses in art, graphic art, theatre, music, and engineering. This examination led to the identification of two primary commonalities: (1) the courses stressed following a design process that included preliminary sketches of the design, and (2) the courses included peer and instructor critiques of the proposed designs as a critical part of the learning experience. These ideas provided the basic structure of the proposed course. The specific content for the course was developed by examining literature on interface design. This examination identified four specific areas of interface design that needed to be addressed in the course: (1) design for information display/presentation, (2) design for data entry, (3) design to perform an operation, and (4) design for interaction to achieve an objective. This information provided the basis for the development of the following course objectives.

Upon completion of the course the student will be able to:

1. evaluate a software interface for adherence to good design principles,
2. describe the problems associated with poor design of software interfaces,
3. describe the issues associated with information display in a software interface,
4. design a software interface that facilitates information identification and comprehension,
5. describe the issues associated with data entry in a software interface,
6. design a software interface that facilitates error-free data entry,
7. describe the issues associated with performing an operation in a software application,
8. design a software interface that facilitates the performance of an operation,
9. describe the issues associated with using software to accomplish an objective, and
10. design a software interface that facilitates accomplishment of an objective.

The new course would focus on developing both the theoretical and the practical knowledge required by developers to create effective interfaces. This course would include the study of human-computer interaction, human factors, and the limitations/opportunities presented by the media used to create the software product. This media consists of any I/O device attached to the computer, such as the screen, mouse, keyboard, printer and speakers. The course also would emphasize developing knowledge and applying basic principles of interface design. A significant amount of time is spent developing and critiquing interface designs. Critiquing is an important part of design education. After each assignment is turned in, a class period should be committed to having students critique other students' work. This is a common practice in the arts, but is not common in IS. The purpose of the critique is to help students improve, both by studying other students' work and having their own work evaluated by peers.

Two components of the course are key to its effectiveness. The first is the principle of preliminary design. It is imperative that interface designers consistently use sketches to develop their ideas about how the interface should look and feel before they begin committing to a design by actually putting menus and buttons on a window. The preliminary design must be thought about and tested while options are still open and the cost of change is small (Edel, 1967). The second important component is the critique of the prototype interface by both the professor and other students. Demonstrating failure in design is often the most important learning tool for improving design (Petrosky, 1992). Another important point is that students should be given as much choice as possible in creating their designs. Through this, they develop the skills to make good design choices (Kimbell, 1982). This is especially important because, as Spolsky (2001) points out, interface design is the art of making choices so the user doesn't have to.

THE COURSE AS TAUGHT

The proposed course was taught in the 2006 spring semester at the authors' university. The course was taught as one half of a required three credit Master of Science in Information Systems (MSIS) degree course. The other half of the course was C# programming. The schedule used is presented below. Each week, approximately one-half of the three hour evening class was devoted to user interface design.

Critiques of preliminary designs were done in groups of two. Students exchanged, reviewed and discussed sketches. The instructor circulated through the class, listening to, and participating in, the discussion when needed. Preliminary designs were required to be in handwritten sketch form. Some students wanted to use Visual Basic or some other program immediately to do their preliminary design, but were not allowed to because this would anchor their design prematurely. Early in the class, sketches primarily employed typical interface components such as listboxes and buttons. However, with encouragement, later preliminary sketches were not as restricted, and the students' creativity and growing knowledge of design enabled more interesting ideas. The drawback was, however, if they got too creative, it became difficult to implement the solution in a standard development environment.

Presentation of the final design for each assignment originally was scheduled with two students presenting, the whole class discussing, and two or three students writing critiques of the design. The students found these discussions quite interesting and asked to have more students present their final designs. The class decided that it would be good to have two students present their design in-depth, and then have the rest of the class provide an overview of their final design. Often, however, time management became a problem when students found some of the extra designs to be interesting, and the discussion went too long.

Actual Schedule
The actual course included five assignments and critiques. The actual schedule of the course, without the C# portion, is included below. The original plan was to spend 1.5 hours on interface design and 1.5 hours on C# each evening class. Due to the time management issues noted above, the interface design portion often ended up taking about 2 hours per class.

Week 1	Introduction to User Interface Design: The Human & the Computer	Spolsky Ch. 1, 2
Week 2	Interface Design Principles	Spolsky Ch. 3-8, 14
Week 3	Discussion/Presentation of Evaluated Interfaces	
Due:	**UI Assignment #1**– Evaluate an Interface	
Week 4	Design for information display	Spolsky Ch. 16
Week 5	Design for information display Critique of Preliminary Design	
Due:	**Preliminary Design UI Assignment #2**	
Week 6	Discussion/Presentation of UI Assignment #2	
Due:	**UI Assignment #2** – Design for Data/Information Display	
Week 7	Design for Data Input Critique of Preliminary Design	Spolsky Ch. 9-11, 15
Due:	**Preliminary Design UI Assignment #3**	
Week 8	Discussion/Presentation of UI Assignment #3	
Due:	**UI Assignment #3** – Design for Data Input	
Week 9	Design to Perform a Task	Spolsky Ch. 18
Week 10	Design to Perform a Task Critique of Preliminary Design	
Due:	**Preliminary Design UI Assignment #4**	
Week 11	Discussion/Presentation of UI Assignment #4	
Due:	**UI Assignment #4** – Design to Perform an Operation	
Week 12	Design for Interaction	Spolsky Ch. 12, 13
Week 13	Design for Interaction Critique of Preliminary Design	
Due:	**Preliminary Design UI Assignment #5**	
Week 14	Discussion/Presentation of UI Assignment #5	
Due:	**UI Assignment #5** – Design for Interaction **Exam**	

EVALUATION

The impact of the course was measured with a pre/post test examination of student performance on a limited set of subject matter and with a survey of the student's perception of the course. This evaluation is limited by low enrollment in the course (7 students). However, the findings are informative.

Student Performance
Student performance was examined by evaluating responses to three questions: 1. Explain the importance of a properly designed user interface. 2. Describe at least three principles of good interface design. 3. Explain the process of good interface design. These three questions cover the breadth of the topics in the course. The questions were given to the students as a pretest on the first night of the course and again as part of the final exam on the last night of the course. The pre-test was administered after the instructor and subject were introduced. Students were asked to do their best so the College and the instructor could evaluate the value of the new topic. Students were given 30 minutes complete the pre-test. Students were not informed of their performance on the pre-test.

Although there were no direct incentives for students to do their best on the pre-test, we believe the results are trustworthy. Students at the master's level generally are aware that they have a responsibility for their own learning. Telling them that the results of the pretest would improve the course they were currently enrolled in should be an incentive. In addition, on the first night of class, students generally are motivated to do a good job to not disappoint the instructor.

The pre-test questions were approximately 30% of the final exam. They were not identified as pre-test questions, nor were students reminded that the final exam results would be compared to pre-test results. Students were given 90 minutes to complete the entire final exam.

The three pre-test and post-test questions were graded on a ten-point scale by a graduate assistant and another professor. Student names were removed from both pre- and post-test questions, and pre- and post-test questions were mixed together and not identified as pre- or post-test. Both the graduate assistant and the professor were given a key prepared by the instructor of the course and were told to evaluate the questions based on the key. Although the sample size is too limited to examine statistically, there generally was agreement between the professor and graduate assistant, however, the graduate assistant generally scored the answers lower than the professor did. A cursory examination of the grading indicated that the professor did more interpretation of the answers than the graduate assistant did. The graduate assistant tended to apply the key more strictly to the answers provided. The average of the professor and graduate assistant scores were used to compare pre/post test results (Table 1).

Question	Pre-Test Score	Post-Test Score	Difference
1. Explain the importance of a properly designed user interface.	5.07	7.86	2.79
2. Describe at least 3 principles of good interface design.	5.21	9.14	3.93
3. Explain the process of good interface design.	5.21	8.07	2.86

Table 1. Average Pre-test/Post-Test scores.

These results suggest that the course did have an impact on the student's knowledge of interface design. Significant gains were made in all three areas. The most interesting, and potentially important, of these is the first question concerning the importance of the interface. This topic was addressed directly only as a portion of the first lecture. However, the increase it showed was as almost as great as the process of design, a topic that was addressed directly or indirectly in almost every lecture. The importance was directly related by a student, who said during the class, "I never thought about it before, but with our new ERP system, the job that I do takes about 5 minutes to complete. I could do it in about 1 minute with our old system." This is an important outcome because the students in the MSIS program are either managers of software development or striving to be managers of development. Getting them to understand that the interface is not an afterthought and should be a major part of the system design could have a major impact on the usability of the organization's systems.

Student Perceptions
Student perceptions of the course were collected to determine if students saw value in the course and how the course could be modified to better serve the students. Ten questions were asked as part of the University sponsored student evaluation. Two seven point Likert scales were used for the responses. One scale focused on agreement (disagreement) with a statement, and the other scale focused on assessment of how much (little) time was spent in particular areas of the course. Responses were anonymous. Results are presented in Table 2.

Based on their responses, students strongly believed that the course topic was very valuable. Only two students did not Strongly Agree (7) that the topic was valuable. Those two responded with the slightly less strong response of 6. The results of the performance question in the previous section, and discussions during the class indicate that the students, through the course, recognized that the software's interface had a big impact on business and that something could be done about it. Interfaces did not just have to be "accepted." The survey also gave students a chance to comment on the course, and three of the seven students specifically noted that the topic was very important to both managers and programmers. Students perceived the depth and breadth of coverage of the topic to be about right.

Students found that the design assignments, critiques of preliminary designs, and presentation of final designs were very valuable in learning the course material. They did not see as much value in writing critiques of each other. Students' lack of support for written critiques is probably because much discussion/critiquing took place during presentations, so writing the critique added little to their learning.

Students found time management in the course to be adequate. They felt that the amount of time spent lecturing and critiquing preliminary design sketches to be about right. However, they were less certain about the amount of time presenting and critiquing final designs. This is interesting because for the first two assignments, only two students presented their final designs to the class. The students said

they found this to be very valuable and wanted to see everyone's final design. This was done for the final three assignments.

Question	Average	Deviation
1. The topic User Interface Design was a valuable course topic (1 strongly disagree → 7 strongly agree).	6.5	0.65
2. The depth of coverage of the User Interface Design topic in the course was: (1 way too little → 4 about right→ 7 way too much).	4.29	0.95
3. The breadth of the User Interface Design topic in the course was: (1 way too little → 4 about right → 7 way too much).	4.14	1.07
4. The User Interface Design assignments were valuable in developing an understanding the course material (1 strongly disagree → 7 strongly agree).	6.00	0.82
5. Critique of the preliminary sketches was valuable in developing and understanding of the course material (1 strongly disagree → 7 strongly agree).	6.43	0.79
6. Student presentations of their final designs were valuable in developing an understanding of the course material (1 strongly disagree → 7 strongly agree).	6.14	0.69
7. Writing critiques of other students' designs was valuable in developing an understanding of the course material (1 strongly disagree → 7 strongly agree).	4.86	1.46
8. The amount of time used in the course to critique sketches was approximately 5 hours spread over 4 nights. This amount of time was (1 way too little → 4 about right → 7 way too much).	4.36	0.63
9. The amount of time used in the course devoted to presentation of final design and critiquing that design was approximately 7.5 hours spread over 5 nights. This amount of time was (1 way too little → 4 about right → 7 way too much).	5.00	1.29
10. The amount of time spent on lecturing about User Interface design topics was approximately 12 hours spread over 9 nights. This amount of time was (1 way too little → 4 about right → 7 way too much).	3.57	0.53

Table 2. Student Perceptions of the Course.

DISCUSSION AND CONCLUSION

As we have demonstrated by teaching a portion of the proposed course, using sketches for preliminary design and design critiques were highly successful, and effective in teaching students proper design techniques and principles. However, improvements could be made. Several students commented that they would have liked to spend more time critiquing actual interfaces (this was done only in the first assignment). This could be done in association with more written critiques, which students found less useful when done in association with a classroom discussion. Students also found the Spolsky (2001) book to be a little light. Using the Cooper and Reitman (2003) book would probably alleviate this problem.

Interface design needs to be included more prominently in the IS curriculum. Our experience teaching the course suggests that even students who have worked in the IS field had not learned or even known about many of the components of human/computer interaction that contribute to developing an effective interface. They also had never considered the business impact of the user interface. Student perceptions and comments from the course indicate that they feel that the course is very important to both managers and programmers and needs to be taught more thoroughly in the IS curriculum.

As this stream of research is at its inception, there are several interesting and promising avenues of further study. Possible topics for research include examining how companies who keep GUI design and development in house, but outsource the rest of the development, are able to achieve a fully integrated solution. We expect to see that the use of proper object-oriented development techniques will make it simple to perform this integration. A more focused topic is to examine the use of the proposed peer

critique technique. It would be interesting to find how well this technique works in this course as well as in other IS courses where it might be appropriate.

REFERENCES

Alanis, M., Mendoza, A., and Alanis, D. "The Offshore Outsourcing Market: The Impact of Client's Perceptions and Supplier's Beliefs on the IS Curriculum'" *Proceedings of the 2004 International Conference on Informatics Education Research*, Washington, D.C. 2004, pp.183-190.

Carmel, E., and Agarwal, R. "Tactical Approaches for Alleviating Distance in Global Software Development," *IEEE Software*, (18:2) 2001, pp. 22-29.

Edel, D.H. *Introduction to Creative Design*, Prentice-Hall, Englewood Cliffs, NJ. 1967.

Galitz, W. O. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*, John Wiley & Sons, Inc. New York. 2002.

Kimbell, R. *Design Education: The Foundation Years*, Routledge & Kegan Paul, London, 1982.

Marcus, A. "Return on Investment for Usable User-Interface Design: Examples and Statistics," *AM+A White Paper* Aaron Marcus & Associates, New York, 2002.

Moitra, D. "India's Software Industry," *IEEE Software* (18:1), 2001, pp. 77-80,

Norman, D. A. *The Design of Everyday Things*, Basic Books, New York, 2002.

Perez, J. C. Gartner: "Offshore Outsourcing Gains Steam*," ComputerWorld*, January 30, 2003.

Petrosky, H. *To Engineer Is Human: The Role of Failure in Successful Desig*n, Vintage Books, New York, 1992.

Petrosky, H. *Small Things Considered: Why There is No Perfect Design*, Alfred A. Knopf, New York, 2003.

Raskin, J. "Looking for a Humane Interface: Will Computers Ever Become Easy to Use?" *Communications of the ACM* (40:2), 1997, pp. 98-101.

Seffah, A.,and Metzker, E. "The Obstacles and Myths of Usability and Software Engineering," *Communications of the ACM* (47:12), 2004, pp. 71-76.

Spolsky, J. *User Interface Design for Programmers*. Apress, Berkeley, CA, 2001.

Swanson, G. "Graphic Design as a Liberal Art: Design and Knowledge in the University and the 'Real World,'" in *The Education of a Graphic Designer*, S. Heller, ed.. Allworth Press, New York, 1998, pp. 13-24.