Software Developers' Attitudes toward User-Centered Design

Theodore Frick, Ph.D.
Elizabeth Boling
Kyong-Jee Kim
Daniel Oswald
Dept. of Instructional Systems Technology
School of Education
Indiana University, Bloomington, IN 47405

William Sugar, Ph.D.
Dept. of Librarianship, Educational Technology & Distance Instruction
East Carolina University
Greenville, NC 27858

Todd Zazelenchuk
Dept. of Instructional Systems Technology
School of Education
Indiana University, Bloomington, IN 47405

Abstract

The concepts of usability and user-centered design (UCD) have grown in popularity over the past 20 years as measured by the number of research and mainstream articles devoted to their discussion. As with all new developments, however, there is always the question of how things work in practice compared to theory. A survey on 83 software developers mostly in small-to-medium sized companies in a variety of industries was conducted to examine software developers' views on UCD and usability practices and to illuminate how current practices relate to theory. Results of a descriptive analysis of the 22 Likert-scale attitude question items suggested that respondents had moderately positive attitudes towards UCD activities and discipline. The Likert-scale items were subsequently factor-analyzed and the results suggested that the respondents tended to agree that UCD is worth the effort and cost. They also tended to agree that it is important to conduct many use test sessions and they learned a lot about their products from user test sessions. Software developers who reported that their companies followed important UCD practices were more likely to agree with the view that UCD is worth the effort and cost. Those who have attended usability test sessions were more likely to agree that user test sessions are valuable, and that UCD is worth the effort and cost. However, those who have attended usability test sessions also were more likely to agree that UCD is more work and costs more than conventional development activities. Also, significantly more good usability practices were reported by software developers who worked on teams that either hired usability consultants or had a usability specialist on their teams compared with those who had no usability specialists at all. While software developers held positive attitudes towards UCD, it was notable that they did not report that their companies used practices that are central to UCD. It appears that, while many software developers agree that UCD is a good idea, it tends not to be implemented fully in practice.

Introduction

The concepts of usability and user-centered design (UCD) have grown in popularity over the recent 20 years as measured by the number of research and mainstream articles devoted to their discussion. As with all new developments, however, there is always the question of how things work in practice compared to theory. The opportunities for UCD to have a significant impact on the rapid developments in information technology are infinite in number and will only increase as new technologies emerge. To ensure that this impact is realized, however, organizations must understand how to best organize themselves and their practices to take full advantage of what UCD has to offer.

The purpose of this study is to examine software developers' views on UCD and usability practices in order to illuminate how current practices relate to theory. The results of this study will help researchers and organizations understand what factors are critical to integrating an effective UCD approach in the development lifecycle.

Defining the Concepts

User-centered design is an approach to product development that emphasizes keeping the end user "front and center" throughout the product development process. Unlike the specific techniques and methods that make it up, UCD is a philosophy toward designing products (Norman, 1988), the underlying theme of which is that developers who stay attuned to the concerns, thought processes, habits, and preferences of the people targeted to use their products will develop interfaces and services that are easier to use, have greater utility, and are more enjoyable for their customers (Rubin, 1994).

If UCD is the philosophy that guides an effective development process, usability may be seen as the end result. Once known simply as "user-friendliness" (Norman & Draper, 1986), the concept of usability has attracted much attention over recent years and is currently considered to consist of the following five attributes: (a) learnability, (b) efficiency, (c) memorability, (d) errors, and (e) satisfaction (Nielsen, 1993).

Perhaps one of the most valuable tools in the designer's UCD toolbox is usability testing. This method affords the design team the unique opportunity to observe the actions of the target user population first-hand. Usability testing allows designers to observe authentic users performing authentic tasks and scenarios. While many tests occur in a laboratory environment to make observation and data collection easier, field visits to the users' actual workplaces provide the additional benefit of an authentic context as well.

Dumas and Redish (2000) identify five characteristics that define usability testing:

1. The primary goal is to improve the usability of a product
2. The participants represent real users
3. The participants perform real tasks
4. You observe and record what participants do and say
5. You analyze the data, diagnose the real problems and make recommended changes to fix those problems (p.22).

The Importance of "Getting Close" to Your Users

The activity of "requirements gathering" has long been a core element of the software design process (Boehm, 1988), yet critics of poorly designed software point out that gathering requirements through focus group discussions or by talking to management often fails to identify what is needed to make a usable product. The only way to accurately define what people will be able to use is to gather information *directly from the users themselves*. As with so many things in life, however, all user-centered design activities are not created equal. Some methods are more successful than others at bringing users and designers close together.

It would seem that the simplest way of gathering information from users is to ask them what they want. Unfortunately, we know that users do not always know what they want. Indeed, Andre & Wickens (1995) cite a host of studies demonstrating that users not only don't know what they want, but that they frequently make bad choices as well. In one study of six different interface designs, users consistently indicated a preference for those designs that they performed most poorly on (Bailey, 1995). The results emphasize how important it is to include empirical data on performances in addition to asking users what they like.

Conducting needs analysis interviews and performing content sorting activities with users have also been found to bring users and designers closer together (Corry, Frick, & Hansen, 1997). These activities have the added benefit of being able to be performed early in the design process, allowing multiple iterations to follow.

Low-fidelity or paper prototyping is a technique that involves users early in the design process (Sugar & Boling, 1995) and has been shown to be just as effective as prototyping exercises that employ a more completed

electronic version (Virzi, Sokolov, & Karos, 1996). The fact that paper prototypes of a computer system interface are obviously unfinished allows users to freely comment and contribute their ideas for improvement to designers (Datz-Kauffold & Henry, 2000).

Testing electronic prototypes or even an fully functioning system has certain advantages over low-fidelity prototyping. On-screen interactions no longer need to be simulated and colors, resolution, modes, and system operating speed can be evaluated more accurately. Misanchuk, Schwier, & Boling (2000) describe how usability testing the working version of an electronic book on instructional multimedia led to the discovery of multiple, desirable features that were missing.

Often, there are factors affecting usability that cannot be observed in a lab or test environment. Contextual inquiry (Beyer & Holtzblatt, 1998) attempts to overcome this problem by having designers observe users in their natural work environment in order to fully consider the many variables that may influence how a product is ultimately used.

Inviting users to participate on the actual design team is another strategy for bringing users and designers close together. Known as participatory design, this approach typically has designers and users work side by side throughout the development cycle. Benefits of participatory design have been shown to include a sense of ownership among users and an increased understanding of users by designers (Williams & Traynor, 1994). Clement & Van den Besselaar (1993) stress, however, that for participatory design efforts to succeed, users must be allowed to take an independent position on problems and they must participate in the process of decision making.

The Current State of Practices and Attitudes

While much has been studied and written regarding usability evaluation methods and design practices, very little work has been done in determining actual practices and attitudes of those in industry. Gould & Lewis (1985) surveyed 447 software developers attending a human factors workshop to see whether they identified three basic principles of designing for usability as a common part of their own design processes. The principles included an early and continual focus on users and tasks, empirical measurement, and iterative design. The results revealed that developers either did not identify with the three principles or did not understand them well enough to implement them as intended.

In another survey of current practices at the time, Dillon, Sweeney, & Maguire (1993) conducted a survey of the software industry in the United Kingdom, gathering data on four themes: respondents' backgrounds, their interpretation and appreciation of the concept of usability, current practice with regard to usability evaluation, and problems and requirements for support in conducting usability evaluations. The authors found a widespread awareness of usability among respondents, but what seemed to be only a superficial application of Human Factors methods.

Differences in attitudes toward usability were considered in a study that combined survey and qualitative interview research of three Management Information System (MIS) managers and 125 end-users of commercial software systems (Morris & Dillon, 1996). Interviews with the managers revealed an emphasis on costs and system features when designing or selecting new technologies for their organizations. This was in significant contrast to users' main concerns of contextual and environmental issues that affect the software's usability.

Methodology

The following questions were considered during the course of this investigation: What are software developers' attitudes toward user-centered design? What are the actual methods utilized by software developers who report using UCD? Does there exist any correlation between the practice of user-centered methods and developers' attitudes?

Data Collection

Three survey forms were sent to each of 500 software companies (1,500 forms). The companies that the survey forms were sent to were selected from the Software Publishers Association membership directory. The survey forms were comprised of questions concerning the respondents job classification (type of software designer), type of training (if any) in usability, history of participation in usability tests, types of usability procedures utilized by respondents, size of the company the respondent worked for, and attitudes concerning usability testing. Most of the questions required respondents to check boxes indicating the appropriate answer, except for the attitude questions which contained 22 question items measuring the subjects' response to a given statement. These items were measured on a five-point Likert scale from strong disagreement to strong agreement.

Subjects

Eighty-three software developers responded to our survey. This was an effective return rate of 5.5 percent.

Results of the Study

Descriptive Analysis for Software Developers' Current Practices of UCD

(1) Respondents' Position as a Software Developer

As shown in Table 1, the majority of respondents worked in commercial applications, instructional software, and entertainment software. Software developers who belonged to the above three positions accounted for 86 percent of the respondents.

Table 1. Software developers' Position

Position	Frequency
Commercial applications	33
Instructional software	24
Entertainment software	14
Other	9
Technical/programming language	1
Instructional and Entertainment	1
Online and Commercial	1
Operating systems	0
Online documentation/help	0
Total	83

(2) Companies that Participated in the Survey

Among the 500 companies to which the mail surveys were sent out, results were returned from 56 different companies. Most of the companies participated in the survey had one person respond; whereas 17 companies surveyed had 2-3 software developers respond.

(3) Number of Employees in the Company

More than half of the respondents (55%) worked in companies with 1-50 employees, and about a quarter of the respondents worked in companies with 51-250 employees. The remaining 18 percent worked in companies with greater than 251 employees. Thus, most of the respondents (82%) worked in smaller companies, with less than 250 employees.

(4) Number of People Assigned to Software Development Team

More than half of the respondents worked in core development teams with 4 - 8 members, and about one-third of the respondents worked on teams of 3 or less. Relatively few respondents (10%) worked on larger teams with more than 8 members.

(5) Company' Expectation on Usability in Their Products

Eighty-two percent of those surveyed reported that their company always expected their software development teams to ensure usability in their products. Seventeen percent indicated that their companies sometimes had such expectations. No respondent indicated that such an expectation did not exist in his or her company.

(6) Use of Specific Process for Incorporating UCD into Product Development

Thirty-seven percent of the respondents answered that their companies always used UCD processes in their product development, and 47 percent responded their companies used UCD processes sometimes. Meanwhile, 14 percent indicated their companies never had such a process in their software development. Thus, it can be seen that the majority of the respondents reported that their companies used specific UCD processes for software development.

(7) Experience in UCD Practice for Software Development

Respondents had an average of approximately 4 years of experience in UCD practice (M= 4.02, SD=1.70), ranging from 0 to over 7 years. More than half of the respondents indicated that they have practiced UCD 1 -7 years.

(8) Experiences in Attending User Test Sessions

For the question asking for their experiences in attending user test sessions, 80 percent of respondents indicated that they had attended at least more than one user test session. In contrast, 20 percent responded that they had never attended user test sessions. There was considerable dispersion in the reported number of times that they attended user test sessions, ranging from 1 to 100 test sessions.

(9) Experiences in Helping Conduct User Test Session

Sixty-four percent indicated that they helped conduct a user test session more than once, while 36 percent answered they never helped conduct user test sessions. Those who had conducted user test sessions themselves reported widely dispersed occurrences, ranging between 1 and 200 times.

(10) Software Development Activities Being Practiced

Respondents were asked to check all of the kinds of activities in which their companies were engaged in developing their products. These activities were ranked according to how frequently they were mentioned by the respondents in Table 2. "Using common sense" was most frequently checked (77), while "using constructive interaction techniques" was least frequently indicated (18). These activities were coded with +, 0, and – by the investigators (the respondents did not see these codes). A "+" activity means that the user is actively involved in the software development process. A "–" activity indicates that the user is not involved in the process. A "0" activity means a user is somewhat involved or neutral. The median split from Table 2 reveals that the majority of the activities above the median do not involve users in the design process (7 out of 9), and the 4 activities in which users actively participated ranked among the bottom 6 (below the median).

Table 2. Software Development Activities Reported to Be Practiced

Code*	Development Activities	Number of Responses
-	Using common sense	77 (top ranking)
-	Setting major goals	67
0	Using computer prototypes	65
-	Interviewing representative users and asking whether they like the product	63
-	Soliciting feedback from "seed sites" or "beta-testers"	62
-	Performing competitive analyses of competing products	58
0	Testing out major design issues with users	58
-	Following GUI guidelines	58
-	Following standard interface guidelines	56
-	Passing screen shots to other developers	50 (median ranking)
0	Using paper prototypes	44
-	Following company's interface guidelines	43
-	Expert walkthroughs	39
+	Doing field studies/visits of user's work environments	38
+	Having a real user on the design team	29
0	Performing task analysis of user's tasks	27
0	Recording user's actions with a program	25
+	Using think-aloud protocols	22
+	Using constructive interaction techniques	18 (lowest ranking)

* + indicates the user is actively involved in the software development process.
- indicates the user is not involved in the software development process.
0 indicates a user is somewhat involved or neutral to the development process.

(11) Attending Formal Training in Usability and Sources of Training

Less than one-third of the respondents indicated that they had any formal training in usability; the remaining 70 percent reported not having attended any formal training in usability. When asked their sources for

obtaining training in usability – including informal types of training – the respondents were most likely to get their training from books/journals, and conferences/workshops.

(12) Accessibility to a Usability Specialist

Sixty percent of those surveyed answered that their development teams did not have access to usability specialists. On the other hand, 17 percent had access to usability consultants, and 17 percent had a specialist on their team; thus, one-third of respondents indicated having access to usability specialists.

Software Developers' Attitudes Towards UCD

(1) Factor Analysis

A factor analysis of the 22 Likert-scale attitude questions was conducted using the image factoring method with varimax rotation, resulting in seven factors as shown in Table 3. In addition, a reliability analysis on each factor was conducted (Cronbach's α for internal consistency). The Likert-scale values of the items that had negative loadings on a factor were reversed when factor scores were computed. Results of the reliability analyses ranged from .5952 to .7784 (see Table 3).

Table 3. Results of Factor Analysis and Reliability Analysis

Factor		Items	Reliability coefficient
1.	User-centered design (UCD) is more work and costs more.	16. My team's UCD activities tend to lengthen development time for our product. 5. The UCD activities that I have participated in did not generally add time to product development cycle.* 13. UCD is more expensive than traditional product development. 1. UCD activities make extra work for me as a developer.	α = .7784 (n = 74)
2.	User test sessions are valuable.	14. Participating in user test sessions is a positive experience. 21. Overall, I do not enjoy participating in user test sessions.* 18. I usually have confidence in the results of user test sessions.	α = .7554 (n = 74)
3.	Epiphany: experience of UCD changed my mind.	2. Users in the test lab behave just the way I expected them to before I started attending user test sessions.* 15. Once I became involved with UCD activities, I changed my mind about what UCD is. 17. After the first user test session I observed, I found that I had an altered view on my users.	α = .6438 (n = 59)
4.	Learned little from user tests and UCD.	3. User test sessions usually do not give me new insights about my program. 11. Participating in UCD activities had little effect on my understanding of this discipline.	α = .5952 (n = 67)
5.	Many tests important, learned a lot about my product.	10. It is important to conduct user test sessions many times throughout product development. 12. I usually learn a lot about my product as a result of user test sessions.	α = .605 (n = 74)
6.	UCD is worth the effort and cost.	8. In general, I would not recommend that other development teams spend effort on UCD activities.* 6. In my opinion, UCD activities are worth the effort. 7. The expenses incurred by UCD activities are offset by savings elsewhere in the development process or life-cycle of the product. 9. In my development work, I find that the extra time it takes for UCD activities does not frequently enhance my products.*	α = .574 (n = 77)
7.	Usability specialists are helpful in improving product.	19. Most usability specialists are primarily interested in improving the overall quality of my program. 4, Usability specialists do not do much except point out the	α = .641 (n = 57)

"mistakes" of my programs.*

*****These items negatively loaded on a factor, and were reverse-coded when computing scale scores for each factor.

(2) General Attitude

Results of a descriptive analysis of the Likert-scale attitude questions suggest that the respondents had moderately positive attitude towards UCD activities (1=strongly disagree, 2=disagree, 3=undecided, 4=agree, 5=strongly agree). As seen in Table 4, the respondents tend to strongly agree that UCD is worth the effort and cost (M= 4.38, SD= .48). They also tend to strongly agree that it is important to conduct many user test sessions and that they learned a lot about their products from user test sessions (M= 4.11, SD= .68). Likewise, respondents tend to disagree that they learned little from user test sessions and UCD (M= 1.77, SD= .66). The overall mean was 3.79, which suggests moderately positive attitude towards usability.

Table 4. Descriptive Statistics for the UCD Attitude Factors

Factor	N	Minimum	Maximum	Mean	SD
1. User-centered design (UCD) is more work and costs more.	74	1.25	5.00	3.15	.94
2. User test sessions are valuable.	74	2.00	5.00	3.90	.61
3.Eiphany: experience of UCD changed my mind.	59	2.00	5.00	3.51	.69
4. Learned little from user tests and UCD.	67	1.00	4.50	1.77	.66
5. Many tests important, learned a lot about my product.	74	2.00	5.00	4.11	.68
6. UCD is worth the effort and cost.	77	3.00	5.00	4.38	.48
7. Usability specialists are helpful in improving product.	57	2.00	5.00	3.95	.78
Overall attitude	45	3.05	4.91	3.79	.37

(3) UCD Practices and Attitudes

Point-biserial correlations were done between the user-active UCD methods (presented in Table 2 with the + codes) and the 7 components of user attitudes toward UCD (presented in Tables 3 and 4). None of these correlations was significant at the 0.05 level. A new variable, "good UCD practices," was then constructed by totaling the number of items checked from the following practices: 1) testing out major design issues with users, 2) doing field studies/visits of users' work environment, 3) using paper prototypes, 4) using computer prototypes, 5) using think-aloud protocols, and 6) recording users' actions with a program. A significant positive correlation was found between the number of "good UCD practices" and the software developer attitude that "UCD is worth the effort and cost" ($r = .235$, $p < .05$). However, none of the other six UCD attitudes was significantly correlated with the number of "good UCD practices," nor was the overall attitude towards UCD correlated with the number of "good UCD practices."

(4) Developers' Background and their Attitudes

1. Position and Company Size

An analysis of variance (ANOVA) was conducted to see if there were any differences in UCD attitudes according to software developer position. Results showed that instructional software developers more strongly agreed that "user test sessions are valuable" than commercial applications software developers ($F = 6.874$, $p < .002$). Furthermore, instructional software developers tended to more strongly *disagree* that they learned little from user test sessions and UCD than did entertainment software developers ($F = 3.618$, $p < .033$). However, no significant difference was found among the software developer positions and their overall attitude towards usability ($F = 2.471$, $p = .098$). Also, there was no difference in attitudes towards UCD according to size of company ($F = 1.164$, $p = .341$).

2. Formal Training in Usability

Results of an ANOVA showed that there was no significant difference in the respondents' attitudes towards usability between the group who have had any formal training in usability and those who have not received any formal training in usability ($F = .213$, $p = .647$).

3. Experiences in User Test Sessions

Results of an ANOVA revealed that those who have attended user test sessions tend to more strongly agree that "user test sessions are valuable" ($F = 3.934$, $p= .051$), and that "it is important to conduct many user test sessions and they have learned a lot about their products from the user test sessions" ($F = 3.337$, $p = .072$). However, those who have attended user test sessions also tend to more strongly agree that UCD is "more work and costs more" than conventional development activities ($F = 4.634$, $p = .035$). In addition, those who have experience in helping conduct the user test session also tend to more strongly agree that "UCD is worth the effort and cost" ($F = 5.555$, $p = .021$) than those who have no such experience.

4. Access to Usability Specialists and the Developer's "Good UCD Practices

An ANOVA was conducted between the group who had a usability specialist either in their team or as a consultant and those who had no access to a usability specialist in order to compare the number of "good UCD practices" between those two groups. The results showed that significantly more good UCD practices were reported by software developers who worked on teams that either hired usability consultants or had a usability specialist on their teams compared with those who had no usability specialists at all ($F = 10.047$, $p = .002$). The means for each group was 3.9 for the group who had access to usability specialists, and 2.7 for the group who had no such access, respectively.

Discussion

Results suggested that while the respondents considered the UCD process more work and additional cost, they viewed UCD as a positive, worthwhile practice. The software developers who attended user test sessions and helped conduct sessions reported more positive attitudes than others. This would suggest that active participation in usability tests may be a factor in developers' positive outlook concerning usability tests.

We were intrigued, however, with the apparent lack of findings in many of the areas for which we performed analyses. For example, we had expected attitude differences between those who have had formal training in UCD and those who have not. No difference in attitude was found.

Most intriguing, however, was the software developers' apparent lack of use "good UCD practices." That is, while the respondents reported UCD as a positive and beneficial practice, this was not correlated with reported numbers of good UCD practices.

Perhaps most important is having a usability specialist either as a consultant or as a team member, since this is associated with greater numbers of good UCD practices. Given that only 30 percent of software developers have received any formal training in usability -- and when they did, it was most likely from books and journals -- it appears worthwhile to have usability specialists on the development team.

Finally, this study is limited by the generalizability of its findings. We do not know if those who responded to the survey are representative of software developers in general. Also, our data were collected in 1994-96, and it is possible that software developers' attitudes and their company's UCD practices may have changed since then. However, our results seem to be consistent, in part, with a recent survey of HCI professionals in North American industry in which respondents were asked to identify what organizational approaches and usability methodologies they perceived to be most effective in having a strategic impact on corporate decision-making (Rosenbaum, Rohn, & Humburg, 2000). The size and type of company along with the size of the usability group within it were all considered, but no statistically significant relationships proved to exist between the demographic data and the organizational approaches and usability methods employed. In our study, we found no relationship between software developers' attitudes toward user-centered design and good UCD practices.

References

Andre, A. D., & Wickens, C. D. (1995). When users want what's not best for them. Ergonomics in Design, October, 37-41.

Bailey, G. (1995). Performance vs. preference. Paper presented at the Conference of the Human Factors and Ergonomics Society, New York, NY.

Beyer, J., & Holtzblatt, K. (1998). Contextual design: Defining customer-centered systems. San Fransisco, CA: Morgan Kaufmann.

Boehm, B. (1988). A spiral model of software development and enhancement. IEEE Computer, 21(2), 61-72.

Clement, A., & Van den Besselaar, P. (1993). A retrospective look at PD projects. Communications of the ACM: Special issue on participatory design, 36(4), 29-37.

Corry, M., Frick, T., & Hansen, L. (1997). User-centered design and usability testing of a web site: An illustrative case study. Educational Technology Research and Development, 45(4), 65-76.

Datz-Kauffold, B., & Henry, S. L. (2000, August 14-18). Waving magic wands: Interactin techniques to improve usability testing low-fidelity prototypes. Paper presented at the Usability Professionals' Association, Asheville, NC.

Dillon, A., Sweeney, M., & Maguire, M. (1993). A survey of usability engineering with the European IT industry - Current practice and needs. In Alty, Diaper, & Guest (Eds.), People and Computers VIII, HCI'93 . Cambridge: Cambridge University Press.

Dumas, J. S., & Redish, J. C. (2000). A practical guide to usability testing. Norwood, NJ: Ablex.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. Communications of the ACM, 28(3), 300-311.

Misanchuk, E. R., Schwier, R. A., & Boling, E. (2000). Discourse on designing an illustrated hypermedia book. Journal of Educational Multimedia and Hypermedia, 9(3), 223-251.

Morris, M. G., & Dillon, A. P. (1996). The importance of usability in the establishment of organizational software standards for end user computing. The International Journal of Human-Computer Studies, 45, 243-258.

Nielsen, J. (1993). Usability engineering. Cambridge, MA: Academic Press.

Norman, D. A. (1988). The design of everyday things. New York, NY: Doubleday Currency.

Norman, D. A., & Draper, S. W. (1986). Introduction. In D. A. Norman & S. W. Draper (Eds.), User centered system design: New perspectives on human-computer interaction (pp. 1-6). Hillsdale, NJ: Lawrence Erlbaum.

Rosenbaum, S., Rohn, J. A., & Humburg, J. (2000). A toolkit for strategic usability: Results from workshops, panels, and surveys. Paper presented at the ACM CHI 2000 Conference on Human Factors in Computing Systems, The Hague, Netherlands.

Rubin, J. (1994). Handbook of usability testing: How to plan, design, and conduct effective tests. New York: John Wiley & Sons, Inc.

Sugar, W. A., & Boling, E. (1995). User-centered innovation: A model for early usability testing (ERIC Document Reproduction Service No. ED 383 340).

Virzi, R., Sokolov, J., & Karos, D. (1996). Usability problem identification using both low and high fidelity prototypes. Paper presented at the ACM CHI'96 Conference on Human Factors in Computing Systems, Vancouver, BC, Canada.

Williams, M. G., & Traynor, C. (1994). Participatory design of educational software (ERIC Document Reproduction Service No. ED 396 696).