

Usability over Time

Valerie Mendoza and David G. Novick
Department of Computer Science
The University of Texas at El Paso
El Paso, TX 79968-0518
+1 915-747-5725

valmen1032@aol.com, novick@utep.edu

ABSTRACT

Testing of usability could perhaps be more accurately described as testing of learnability. We know more about the problems of novice users than we know of the problems of experienced users. To understand how these problems differ, and to understand how usability problems change as users change from novice to experienced, we conducted a longitudinal study of usability among middle-school teachers creating Web sites. The study looked at the use both the use of documentation and the underlying software, tracking the causes and extent of user frustration over eight weeks. We validated a categorization scheme for frustration episodes. We found that over the eight weeks the level of frustration dropped, the distribution of causes of frustration changed, and the users' responses to frustration episodes changed. These results suggest that the sorts of errors that are most prominently featured in conventional usability testing are likely of little consequence over longer periods of time.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *Evaluation/methodology, Training, help, and documentation.*

General Terms

Documentation, Human Factors, Measurement

Keywords

Usability, training.

1. INTRODUCTION

Testing of usability might perhaps be more accurately described as testing of learnability. We know more about the problems of novice users than we know of the problems of experienced users. And where researchers have looked at more experienced users, the results tend to be “snapshots” of usability rather than longitudinal examination of changes in usability over time. Recent studies of usability and frustration have contributed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGDOC '05, September 21 - 23, 2005, Coventry, UK.
Copyright 2005 ACM 1-58113-000-0/00/0000...\$5.00.

taxonomic analyses but have not significantly addressed the issue of time and experience.

To understand how usability problems and frustrations change as people develop from novice to experienced users, we conducted a longitudinal study of usability, tracking changes in the level and nature of users' frustration over time. In particular, the study looked at the experiences of middle school teachers creating Web pages with a software package that was new to them. The main issues in which we interested included:

- Do users' levels of frustration caused by usability problems change as a function of experience with an application?
- Do the kinds of usability problems users encounter with a new system change over time as a function of use?
- Does the way users respond to usability problems change over time?

To answer these questions, this paper reviews related research, particularly with respect to the causes and measurement of frustration of users of computing systems; explains the study's methodology, including a characterization of the participants, a description of the application domain and the task set, and a presentation of the experimental design; presents the study's results; and briefly discusses limitations and future work.

2. RELATED WORK

Looking at usability for periods longer than initial use requires understanding of the nature of and reasons for users' frustration in using computer systems. In this section we review the literature on understanding and measuring user frustration, and on the kinds of usability problems that cause frustration. These causes can include user inexperience, system complexity, time delays, and poor interface design.

2.1 Frustration

Freud wrote that frustration occurs when a situation hinders or stops someone from reaching their goal. Thus frustrations will occur when a user believes an outcome is incorrect regardless of whether it is their own error or the fault of something else [13, 12].

Studies of usability have tracked frustration as a measure. For example, Bessiere et al. [2] examined the root causes of user frustrations by looking at surveys of 108 users who had worked on a task for one hour. The study found that as the amount of time lost using a computer increased so did the level of frustration. The study also found that computer experience was not significantly related to the amount of frustration experienced.

In contrast, Hazlett [9] found that novice users encountered more frustrating experiences than experienced users. The reason for this difference is that Bessiere et al. measured frustrations only for a one hour time period instead of over the extended time studied by Hazlett. This suggests that deeper understanding of the causes of usability problems, and users' reactions to these, requires going beyond learnability to study use over much longer period of time, perhaps weeks or months.

Ceaparu et al. [7] asked 59 participants to spend an hour on the computer and then report their frustrating experiences. The participants were not given a specific task but were asked to carry out tasks they did every day. The study examined the frequency, cause and severity of frustrating experiences, and time lost due to frustrating experiences. The results showed that users regularly encountered frustrating experiences and that their frustration levels were extremely high. However, the study did not examine frustrations over time with a particular application or consider if there was a correlation between incidents of frustration and perceived levels of proficiency.

2.2 Causes of Frustration

Causes of user frustration identified in the literature include user inexperience, system complexity, time delays, and poor interface design. These causes are interrelated, in that complexity and poor design are likely to be especially troublesome for the novice user. While unfamiliarity may be a problem even for well designed interfaces, poor design may significantly increase the chances that a user will become confused and frustrated. Distinguishing issues of complexity and poor design from issues of inexperience becomes problematic in typical studies of usability, which tend to be short-term studies that find issues of learnability rather than actual usability. In short-term studies, the data resulting from the difficulties of being a novice user may swamp the data attributable to longer-term problems such as complexity, delay, and poor design. Or if the longer-term problems are observed, the data may not permit analysis of changes in usability over time.

2.2.1 Errors of Novice Users

Novice users of software prefer a set of simple actions, but as their experience increases so does their desire for more extensive functionality and rapid performance [19]. But because they, as novices, lack knowledge of the system they are using, they are prone to encounter errors. As they begin using a system, novice users will find many aspects of the system to be frustrating [15, 13]. Users often become frustrated and confused by errors they make in the early stages of learning [6]. Novice users encounter many errors and spend a large portion of their time trying to recover [13].

An error has been defined as a planned sequence of actions that fails to achieve its intended outcome or goal, as long as these results are not attributed to some chance agency that is of no fault of the user or the design [17]. Errors can be seen as comprising three different types: mistakes, slips, and situational errors [12, 15, 17].

A *mistake* is when the user chooses the wrong command for the required task [13, 11, 15, 17]. This type of error is difficult to detect because the action is appropriate for the goal; the problem, however, is that the wrong goal is formed [15].

A *slip* is when the user chooses the correct command to carry out the task, but an error occurs with the execution of the command

[15, 17]. This type of error could occur from a misspelled word or an incorrect sequence of actions. An example of a slip might be where a user saves a file onto the computer's hard drive instead of onto a floppy disk. This is considered a slip because the goal is correct, but the execution of the goal is incorrect.

A third type of error, which developed with the rise of network-based computing, is called *situational* error [13]. This type of error occurs when an individual is using a network that is not functioning properly or is not available. This error cannot be classified as a mistake because the user chooses the correct command but cannot reach his or her goal. The situational error is also not considered a slip because the user did not use the wrong command or incorrect information. Rather something is wrong with the network [11, 13]. For Web browsing and e-mail applications, situational errors were the greatest cause users' frustrations [7].

The effect of error on performance of novice users can be seen through the success of "training wheels" interfaces, which simplify the functions available. For example, Carroll and Carrithers [6] compared use of a word processor with and without a training-wheels interface. Users in the training-wheels condition finished their task faster, had fewer frustrations, experienced less confusion, and had fewer errors.

One of the reasons that errors of novice users cause frustration is that novice users do not understand many of the errors that occur [15]. Indeed, especially when faced with a confusing or misleading interface, novice users may not even perceive errors when they occur or believe an error has occurred when nothing actually happened [12]. Worse, alerting users to errors may cause further problems: novice users encountering error messages become confused, dismayed and discouraged [19], and inexperienced users do not know how to handle system crashes, viruses, and dialog boxes [20].

2.2.2 Complexity

Beyond the inexperience and confusion of novice users, frustration can be caused by inherent qualities of the system being used. Complexity, in particular, has been identified as a factor leading to poor usability [1].

While novice users will do better with simpler interfaces [6], as users gain experience they will seek to use the additional features that will let them do more with the system [19]. For example, of the 265 functions available in Microsoft Word, 27 percent of these functions were used and 51 percent of the functions were familiar to the users. While 62 percent of the users said that unfamiliar functions can be annoying and frustrating, fifty percent of them wanted to be able to discover new functions [1].

Unwanted or unperceived features can lead to frustration even for experienced users. For example, auto-formatting contributed to numerous frustrations that could have been eliminated if the feature had been disabled [14].

2.2.3 Delay

Delay is an important element of situational error that affects both novice and experienced users. Users' reactions to time delays are influenced by their expectations, experience and motivation [19].

As time lost increases, the amount of frustration increases [2]. In a data-entry task, for example, slow response times produced a significantly higher frustration rate than did more rapid response

times [18]. The advent of Internet has led to frustrations caused by network delays. Long download times on the Internet were found disruptive and confusing [12]. When browsing a Web site, users are likely to become frustrated and give up if response time is slow. As delays increase, users begin to respond more negatively to the Web site, and experienced users are more sensitive to these delays [3].

2.2.4 *Poorly Crafted Interfaces*

While well-crafted interfaces assist users and enable them to be productive, poor interfaces lead to reduced productivity, greater frustration, and more errors [11]. Novice users can be confused by “feature explosion” for applications as diverse as word processors [6] and Web browsers [12].

The literature is less helpful with respect to the effect of poor interface design on the performance of more experienced users. Usability testing has been shown effective for users experienced in an application domain when using a novel interface [16], but this study was short-term. A three-year study of the usability of a text editor [8] found that while users continued to explore new functions even after 140 weeks, the users had explored 75 percent of the functions within the first two weeks. This suggests that an observational period of two months should more than adequately disclose usability problems beyond those encountered by novice users. In terms of classifying usability problems, the authors of the study discussed its results in ways highly specific to text editors; higher-level classification of usability issues was not addressed.

3. METHODOLOGY

To study possible changes over time in the causes and nature of frustration with software and its documentation, we used a longitudinal within-subjects experimental design. In this section, we describe the participants, the application domain, the task set, and the design of the study.

3.1 Participants

The study’s participants were faculty from Morehead Middle School in El Paso, Texas. The subjects were not compensated for their participation, as the tasks we studied were part of their regular duties, proficiency with technology is considered in faculty job evaluations, and the school district provided an incentive for the school as a whole. The study’s critical data were collected through reports that were not part of participants’ official duties, so to this extent the study’s success depended on the goodwill of the faculty. Of the 48 teachers who completed a pre-study questionnaire, 32 provided reports for all eight weeks of the study; these responses constituted the principal data on which our conclusions are based. There were 25 females 7 males. Their average age was 43, with a standard deviation of 10. All participants had a bachelors degree or higher. Most of the participants in this study had a computer in their classroom and had Internet access. Of the 32 teachers participating in the study, 5 judged themselves inexperienced or very inexperienced with computers, 16 somewhat experienced and 9 experienced or very experienced.

3.2 Application Domain

Since 1995, the El Paso Independent School District (EPISD) has been subject to United States federal and Texas state mandates to increase the use of information technology in the schools. In the

fall of 2002 the State Board for Educators Certification approved new standards of knowledge of technology for all beginning educators. One of these standards requires teachers to communicate information in a variety of formats. EPISD is complying with this standard by purchasing software that facilitates communication with diverse audiences. EPISD has also purchased software to help educators implement technology into their curriculum. In particular, the IBM Learning Village was one of the software packages the district adopted this year. It enables a school to communicate and collaborate with faculty, staff, parents, and students. The Learning Village is available to anyone with Internet access and allows parents and students to view homework assignments, special projects, read teacher evaluations of student progress, view school Web pages, and conduct online private conversations with teachers. The Learning Village software package contains multiple applications, including Registration Directory, Events and School application, Home Page Designer application, Talk at School application, Teachers Lounge, Private Conference application, Team Project application, Strategies application, and the Team List Manager.

The schools were asked to create and maintain school Web pages using this package by the end of the 2005 spring semester. Our study tracked the frustrations when using the Home Page Designer application, which lets users create Web pages to communicate expectations, projects, homework, calendars, study tips, student work, and other information about classes. The application has a template that users fill out to create specialized Web pages. The application is aimed at users who have very little experience designing a Web page. Indeed, users need no prior experience in creating Web pages, nor any knowledge of HTML code. The Home Page Designer does let intermediate and advanced users create a Web pages using HTML. While our study focused on the use of the Home Page Designer, we note that users could potentially encounter difficulties with other applications in the package.

At Morehead Middle School, before our study, fewer than 10 percent of the faculty used the Learning Village. We provided an eight-week structured task set that would introduce the Morehead teachers to the development of Web sites using the Home Page Designer application. Their overall goal was to develop a Web page that parents and students could access from any computer. In the first week, the participants were introduced to the Learning Village and created their own Learning Village account. The participants were asked to register as new users, to log onto the Learning Village Web site, and to examine the Home Page Design software. In the second week, the participants actually began to use the Home Page Designer. The teachers were asked to locate a link in the Learning Village titled “new homepage” and were then given short instructions on the Home Page Designer and began working individually on their own Web pages. In subsequent weeks, the tasks became increasingly advanced. For example, in week four, participants were asked to implement a calendar in their Web page.

3.3 Experimental Design

The study used a design based on participative evaluation [10], in that the data were gathered principally from self-reports rather than from third-party observation as is typically the case in commercial usability studies. The subjects were given a pre-study questionnaire and then asked to note frustration episodes as they

occurred and to report these weekly. There was no control group because there was no experimental manipulation.

3.3.1 Analytical Approach

The study by Ceaparu et al. [7] was the direct inspiration for the study reported here. Ceaparu et al. provided a key approach to understanding the nature of usability by looking empirically at the causes of users' frustrations. However, the study had some significant limitations, raising issues of subjects, inter-rater reliability, use of categories, and time information:

Subjects. The 37 subjects in the pilot study and the 59 subjects in the main study were all undergraduate students majoring in computer science or computer information systems, averaging 22.7 years old ($sd = 3.8$). As Ceaparu et al. pointed out, future work would include looking at frustrations of users in professional workplaces.

Inter-rater reliability. While Ceaparu et al. classified the frustration episodes in their pilot study into five categories (Internet, applications, operating systems, hardware and other), they did not report inter-rater reliability for the classification of the users' frustration reports. An approach with a higher degree of replicability would assess the reliability of the classifications through a statistic such as Cohen's Kappa [4].

Use of the categories. Although the data in the pilot study were used to derive the five categories of frustration episodes and the data in the main study "helped better define" the categories, the categories were used principally as a way of bundling the data for presentation rather than as a basis for analysis. Given the classifications, the data could have been compared across independent factors.

Time information. The subjects in the main study were asked to report on (a) at least three frustrations they experienced when performing common computing tasks and (b) at least three episodes of frustration that they observed occurring to others. The study did not report time-based data, as it apparently did not collect information on changes in usability and frustration over time. There is no reason to expect that users started as novices to the systems for which they reported frustrations, nor that all of a subject's reports concerned a single system, even if the time-based information could be recovered from the data. Consequently, it remained to be seen if usability problems change as a function of the user's experience with the application being evaluated.

We attempted to address these limitations in the design of the study reported here. The subjects are middle school teachers using computer systems as part of their professional duties. Inter-rater reliability was assessed—and thus the categories validated—using Cohen's Kappa. The validated categories were then used as the basis for examining changes in patterns of frustration over time.

3.3.2 Experimental Protocol

The participating teachers completed a pre-study questionnaire before being introduced to the IBM Learning Village package. The questionnaire provided information about the subject group as a whole and provided a capability to scale later frustration reports based on self-assessed levels of anxiety and unhappiness. As it turned out, our subjects were quite happy: no teachers rated themselves unhappy or very unhappy, 2 of the 32 teachers rated

themselves somewhat happy, and the remaining 30 rated themselves happy or very happy.

The teachers were asked to fill out a post-frustration experience survey every time they encountered a frustrating experience while using the Home Page Designer. With the form, participants were asked to rate their frustrations on a Likert scale, discuss what they found frustrating about the experience, and, if they solved the problem, to indicate how. The set of choices for responses to the usability problem was adapted from that of Ceaparu et al. [7].

The teachers were supported by a weekly training session provided by the lead author as part of her work for EPISD. The training sessions took place in a computer lab that has Internet access. There were enough computers for all participants. A digital projector was also used to enable participants to follow along visually. The teachers were trained in small groups, ranging from six to ten members. Participants could ask questions while working on their Web pages. The technology site coordinator also e-mailed all of the schools some simple instructions to get them started. After the initial training, participants had access to the Learning Village's online tutorial.

4. RESULTS

We now turn to the results of the study. We report the classification of the frustration episodes, analyze frustration and proficiency trends over time, and explore patterns of users' actions after each episode.

4.1 Classification of Frustration Episodes

Over the eight weeks of our study, the participants reported 243 frustration episodes. The episodes were classified into five categories adapted from those of Ceaparu et al. [7], who had developed these categories:

- Internet
- Applications
- Operating Systems
- Hardware
- Other

Our adaptations of these categories were motivated by these factors:

- Most of our subjects' frustrations were related to the Home Page Designer application. Hence we differentiated major groups of problems within the general application category. In particular, we distinguished (a) episodes where a feature was hard to find from (b) episodes where the feature was actually not available in the Home Page Designer application.
- Our subjects reported no episodes arising from problems with hardware.
- We combined problems of the browser with those of the Internet. Using the categories of Ceaparu et al., the browser errors would have been classified as application episodes. For our study, which was specifically focused at long-term changes in usability for the Home Page Designer application, we needed to limit the application categories to this specific application, as our users would not have been novice users with respect to browser applications. We note that

the Internet episodes could be considered situational errors.

- We observed that most of the kinds of episodes that Ceaparu et al. would have coded as “other” were instances of the mistakes and slips characteristic of novice errors. Accordingly, our category is called Operator Error.

Based on these considerations, we classified our data using these categories:

- Hard-to-Find Features
- Missing Features
- Operating System
- Internet, Browser
- Operator Error

Once the categories had been developed, every episode was independently classified by each of four coders. Coders other than the authors received a brief training on how to classify the episodes. Using all of the raters’ classifications, we calculated Cohen’s Kappa [4], the preferred statistic for inter-rater reliability, extended to multiple coders [5]. Kappa varies between 0 (no agreement at all) to 1 (perfect agreement). The Kappa value for our classifications was 0.672, which falls in the range of values generally considered to indicate good reliability. This suggests that our categorization can be considered validated as replicable.

4.2 Frustration over Time

Table 1 shows the users’ self-assessments of their proficiency and frustration level. A repeated-measures test indicated that users’ levels of frustration decreased significantly over the eight weeks of the study ($p < .001$). While reported proficiency levels tended to increase, the inverse correlation of proficiency with frustration level was not significant ($p = .324$).

Week	Proficiency Averages per Week	Frustrations Level Averages
1	1.4	3.8
2	1.2	3.9
3	1.6	3.6
4	1.9	2.6
5	2.1	2.2
6	2.3	1.6
7	2.1	0.9
8	3.4	0.8
Overall averages	2	2.425

Table 1. Proficiency and Frustration Level over Time

The causes of the users’ frustrations also changed over time. As shown in Figure 1, based on data presented in Table 2, there were clear trends in the kinds of episodes that led users to report frustration. In particular, the early peak and relatively quick drop-off in frustration episodes caused by user errors suggest that the sorts of errors that are most prominently featured in conventional usability testing are likely of little consequence over periods of time longer than two or three weeks. In effect, this early incidence

of user errors may mask the more serious problems of hard-to-find features that occur in later weeks.

Hard-to-find features. The most visible trend is that of hard-to-find features, which has more episodes than the other factors, peaks in weeks three and four, and then largely tails off. We expect that the rise-and-fall shape for episodes for this factor is due to (a) the increasing demands of the tasks the users were attempting, (b) the users’ development of a base set of known functions, and (c) their increasing facility in finding new functions.

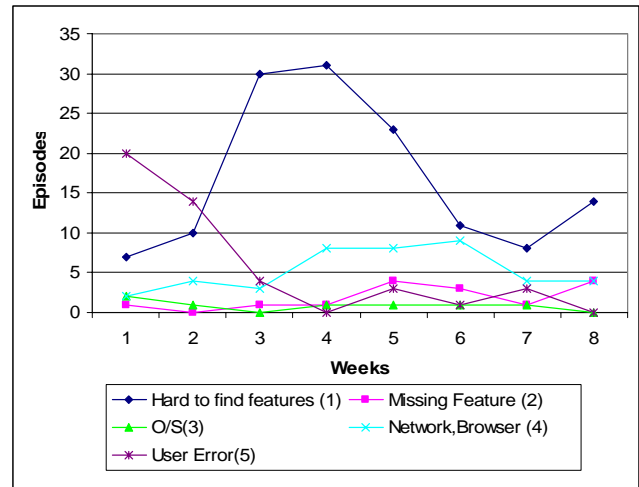


Figure 1. Frustration Episodes over Time

Week	Frustration Episode Category				
	(1)	(2)	(3)	(4)	(5)
	Hard-to-Find Feature	Missing Feature	O/S(3)	Network, Browser	User Error
1	7	1	2	2	20
2	10	0	1	4	14
3	30	1	0	3	4
4	31	1	1	8	0
5	23	4	1	8	3
6	11	3	1	9	1
7	8	1	1	4	3
8	14	4	0	4	0

Table 2. Frustration Episodes over Time

User errors. We had classified novice errors such as slips and mistakes in the general category of user errors. Our analysis of the nature of these episodes seems to be confirmed by their pattern over the eight weeks of the study. The episodes coded as user errors have relatively high levels in weeks one and two, and then fall off to minimal levels for the remainder of the study period. Thus these causes of frustration appear to be truly associated with novice users and represent what one might call “entry barriers” rather than fundamental problems with an application’s usability.

Network and browser. The number of frustration episodes associated with network and browser problems tended to increase gradually, peaking in weeks four, five and six, before declining again in weeks seven and eight. We speculate that this trend was

due to the subjects' increased need for and use of network services and browser functions as they advanced in their use of the Home Page Designer.

Missing features. The trend for missing features was roughly similar to that for Internet and browser episodes. The frustration episodes peaked in weeks five and six. We attribute this pattern to (a) the assignment of tasks in the early weeks for which we knew that the Home Page Designer provided support and (b) the subjects' exploring new features with less-structured tasks as they gained confidence in the later weeks of the study.

Operating system. There were too few episodes of frustration attributable to operating system to discern any trend.

4.3 Users' Responses to Frustration

In the reports of their frustration episodes, the users indicated the action that they subsequently took in response to the problem. The set of possible responses was provided on the report form, so issues of coding and inter-rater reliability for users' responses do not arise.

The incidences of the kinds of user responses, as a percentage of the total per week, are shown in Figure 2. This shows the relative numbers of user responses; in absolute numbers, most of the responses trended down because the number of frustration episodes trends down.

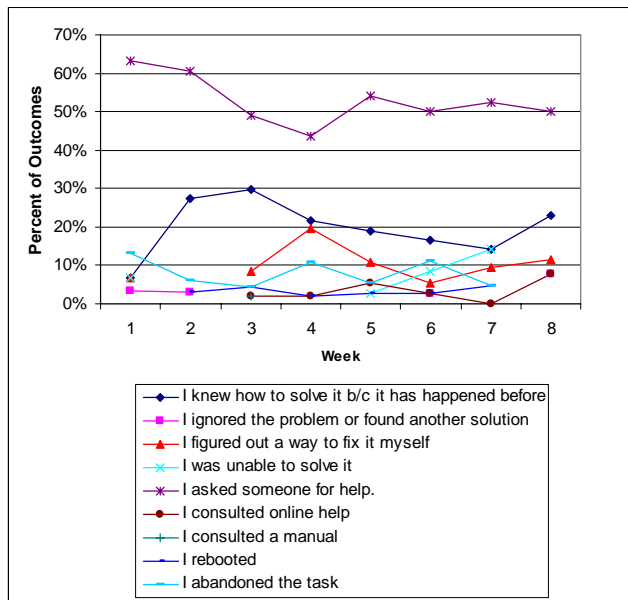


Figure 2. Relative Incidences of Users' Responses to Frustration Episodes

The relative incidences of users who knew how to solve the problem because they solved it before was low in week one (not surprisingly, because they had not had the chance to solve the problem previously) and peaked in weeks two and three. This pattern may be due to the rate at which new features were introduced into the task set.

A similar pattern can be seen for instances where users figured out a way to fix the problem themselves. This kind of outcome was highly infrequent until week three, peaking in week four. We speculate that the reason for this pattern is that the users gained

experience and confidence that enabled them to tackle problems on their own.

Analysis of the response data suggests that abandoning a task was related to frustration level ($p < .01$). Considered with the results on changes in causes of frustration, this suggests that while interface characteristics that lead to high incidences of novice mistakes may lead some users to abandon tasks or applications, conventional usability analysis may focus too heavily on finding and fixing sources of problems that may not be particularly troublesome to users over time.

The aggregated data for user responses to frustration episodes, presented in Table 3, show that by far the most common (52 percent) of user responses was to ask someone else for help. This reflects the fact that, in our study, the trainer and a computer technician were in the lab while the teachers worked on their Web pages. Also, the participants worked in groups of eight to ten people. These teams spanned the entire school year, so the participants had developed relationships with each other, working well together and feeling comfortable with each other.

In 3 percent of the episodes the user responded to the problem by consulting on-line help. In only one instance (0 percent) did a user report consulting a manual.

User's Action after Frustration Episode	N
I knew how to solve it b/c it has happened before	21%
I ignored the problem or found another solution	2%
I figured out a way to fix it myself	9%
I was unable to solve it	3%
I asked someone for help	52%
I consulted online help	3%
I consulted a manual	0%
I rebooted	3%
I abandoned the task	7%

Table 3. Aggregate User Responses to Frustration Episodes

These results contrast with those of Ceaparu et al. [7] in ways that may reflect differences between the user populations and the their tasks. In the Ceaparu et al. study, the users reported that they knew how to solve the problem because it had happened before in about half of the frustration episodes, compared with 21 percent of the episodes in our study. This difference likely reflects the circumstances that the subjects in the Ceaparu et al. study were performing computer tasks that they did every day.

Another striking contrast is that in the Ceaparu et al. study only about 12 percent of the frustration episodes were resolved by asking someone else. In our study, this was the users' actions in over half the episodes. We attribute this difference to the work and teaming environment at Morehead Middle School.

One pattern of responses was consistent across the two studies. In the Ceaparu et al. study, only about 4 percent of the episodes were resolved by using online help and only 1 percent by consulting a manual or book. These results were remarkably consistent with those of our study.

5. CONCLUSION

The data showed a main effect of user frustration dropping over the eight weeks of the study period. We do not know specifically the individual tasks that the users were performing on a session-by-session basis. We expect, though, that the kinds of tasks grew more sophisticated and consequently difficult over the period of the study. This makes the drop in frustration even more marked, and suggests that factors such as features being hard to find and operators committing slips and mistakes really are the principal causes of severe frustration. If there are longer-term causes of frustrations, users may find work-arounds or simply abandon some tasks.

One limitation of our study may be that we may have actually measured repeated learnability rather than usability because we introduced new features and tasks each week. The features learned in previous weeks continued to be used, but we have not separated out reports of frustrations from old or new features. However, this concern may not be a serious problem because the data for the frustration level, categories of frustration episodes, and user responses to frustration all showed changes over time.

Our start in looking at changes in usability over time suggests multiple avenues for future work. Post-training support has been shown to lead to higher retention skills and usage levels for computing tasks [21]. So one interesting line of research would be to correlate the type of training end-users receive and the level of frustration they experience during use of the system. Another line of research would be to correlate the users' subjective reports of frustration levels, especially over time. This would indicate what kinds of usability problems led to greater levels of frustration and whether this relationship changes over time. If a user repeatedly experiences a kind of usability problem over the course of, say, eight weeks, do they have different reactions as a result of their experience? Similarly, are the changes in the distribution of user responses due to changes in users' strategies, or are these simply a result of the changes in the distribution of the usability problems?

More broadly, our results suggest that conventional usability tests catch causes of frustration that represent "entry barriers" for novice users rather than fundamental problems with an application's usability. This result raises questions about the utility of the traditional find-and-fix approach to usability testing. It is certainly a good thing to eliminate the causes of novices' errors, which may lead to abandonment of the application. But conventional usability methodologies may be unlikely to go beyond these kinds of errors to find, much less fix, longer-term sources of frustration, such as problems of hard-to-find features that increasingly frustrated our users as they moved into the middle weeks of the study. While some of the problem of hard-to-find features can be addressed in conventional usability testing through selection of appropriate test cases, we lack the observational or experimental methodologies that would enable developers to see beyond the relatively superficial causes and to detect more serious obstacles to longer-term use without actually having subjects test an application over long periods of time. Development and validation of such methodologies would help address this problem.

6. ACKNOWLEDGMENTS

This work was supported by National Science Foundation Award No. 0080940. Members of UTEP's Interactive Systems group

contributed to the experimental design and evaluation, and the SIGDOC reviewers provided valuable help in focusing the treatment of the results.

7. REFERENCES

- [1] Baecker, R., Booth, K., Jovicic, S., McGrenere, J. and Moore, G. (2000). Reducing the gap between what users know and what they need to know. Proceedings of the ACM 2000 International Conference on Intelligent User Interfaces, 17-23.
- [2] Bessiere, K., Ceaparu, I., Lazar, J., Robinson, J., and Shneiderman, B. (2003). Social and psychological influences on computer user frustration, CS Technical Report 4410, Department of Computer Science. University of Maryland.
- [3] Borella, M.S., Sears, A., and Jacko, J.A. (1997, November). The Effects of internet latency on user perception of information content. *Proceedings of IEEE Global Telecommunications Conferences*, 1932-1936.
- [4] Carletta, J. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2) (1996), 249-254.
- [5] Carlsson, M., Lofstom, L., and Ahlfeldt, H. (2001). Classification of procedures in the domain of thoracic surgery-A study of reliability in coding. *Journal of Medical Systems*, 25(1).
- [6] Carroll, J., and Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM*, 27(8), 800-806.
- [7] Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., and Shneiderman, B. (2004). Determining causes and severity of end-user frustration, *International Journal of Human-Computer Interaction*, 17(3), 333-356.
- [8] Cook R., Kay, J., Ryan, G., and Thomas, R. (1995). A toolkit for appraising the long term usability of a text editor. *Software Quality Journal*, 4(2), 131-154.
- [9] Hazlett, R. (2003). Measurement of user frustration: a biologic approach. *Conference on Human Factors in Computing Systems (CHI 2003)*, 734-735.
- [10] Hilbert, D. (1998). *A survey of computer-aided techniques for extracting usability information from user interface events*, Technical Report UCI-ICS-98-13, Department of Information and Computer Science, University of California at Irvine, March, 1998.
- [11] Lazar, J. & Huang, Y. (2003). Improved error message design in Web browsers. In J. Ratner (ed.). *Human Factors and Web Development* (2nd ed.), 167-182. Mahwah, NJ: Lawrence Erlbaum Associates.
- [12] Lazar J., Meiselwitz, G., and Norcio, A. (2003). Novice user perception of error on the Web. *Universal Access in the Information Society*, 3(3), 202-208.
- [13] Lazar, J., and Norcio, A. (2000). System and training design for end-user error. In S. Clarke & B. Lehaney (Eds.), *Human-Centered Methods in Information Systems: Current Research and Practice*. Hershey, PA: Idea Group Publishing, 76-90.

- [14] Mentis, H. M. & Gay, G. K. (2003). User recalled occurrences of usability errors: Implications on the user experience. *Extended Abstracts of the Conference on Human Factors in Computing Systems, Ft. Lauderdale, Fl*, 736 – 737.
- [15] Norman, D. (1983). Design rules based on analyses of human error. *Communications of the ACM*, 26(4), 254-258.
- [16] Novick, D. (2000). Testing documentation with “low-tech” simulation, *Proceedings of IPCC/SIGDOC 2000*, Cambridge, MA, September, 2000.
- [17] Reason, J. (1990) *Human Error*. Cambridge: University Press, Cambridge
- [18] Schleifer, L., and Amick, B. (1989). System response time and method of pay: Stress effects in computer-based tasks, *International Journal of Human Computer Interaction*, 1(1): 23-39.
- [19] Shneiderman, B. (1998). *Designing the user interface: Strategies for effective human-computer interaction*. 3d ed. Reading, MA: Addison-Wesley.
- [20] Shneiderman, B. (2000). Universal usability: Pushing human-computer interaction research to empower every citizen. *Communications of the ACM*, 43, 5, p.84-91.
- [21] Snoddy, S., and Novick, D. (2004). Post-training support for learning technology, *Proceedings of SIGDOC 2004*, Memphis, TN, October 10-13, 2004.