



Cross-user analysis: Benefits of skill level comparison in usability testing

Laura Faulkner*, David Wick¹

Applied Research Laboratories, The University of Texas at Austin, P.O. Box 8029, Austin, TX 78713-8029, USA

Received 15 April 2004; revised 19 April 2005; accepted 23 April 2005

Available online 5 July 2005

Abstract

This study presents a cross-user usability test approach and analysis technique that extends beyond merely identifying the existence of a usability problem to introducing an empirical basis for identifying the type of usability problem that exists. For experimental purposes, 60 users were tested with three levels of user-competency determined by experience in using: (1) computers, and (2) the tested application. Applying the Tukey honestly significant difference (HSD) test to each test element provided statistical comparison between different experience levels. Analysis results between experience levels suggested which levels encountered usability problems. The authors demonstrate that statistical calculations of cross-user data can render empirical support for categorizing usability problems.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Usability testing; Users; HCI methodology; Usability research; Empirical method

1. Introduction

Often in traditional usability testing, novice testing focuses on learnability while expert testing focuses on optimal use. This difference in focus means different tasks are used and different tests are performed, making them unsuitable for comparison across user levels. This split between novice and expert user testing is placed in historical context by Jakob Nielsen (2000), who traces several shifts in the focus of usability testing. For example, in

* Corresponding author. Tel.: +1 512 835 3328; fax: +1 512 835 3100.

E-mail addresses: laura@arlut.utexas.edu (L. Faulkner), dwick@arlut.utexas.edu (D. Wick).

¹ Tel.: +1 512 835 3646.

the 1970's, when computers were the domain of an elite group, usability was focused on these experts. However, the widespread use of computers by average people in 1980s led to a shift toward novice users, with primary focus on learnability. Referring to the change in focus as a pendulum, Nielsen describes two further shifts one in toward late 1980s to the early 1990s the performance of expert users, and another in the early 1990s arising from web usability issues and focusing on novice users. Nielsen concludes that the current focus on novice testing will start to swing back towards expert testing, and asserts the importance of both forms of testing in providing substantial evidence for usability.

The separate testing of either novice or expert, but not both on the same tests, is supported by examining descriptions of usability tests on various products. One such description discussed the existence of expert users, and implied having received feedback from those experts, but performed testing only on novice users. Further, the description highlighted this marked view of separation by describing information-gathering approaches to fulfill the need 'to understand how each group approaches' the application, with 'each group' specifically being composed of either novices or experts (Benson, 2001; see also Straub, 2003 and Fu et al., 2002). Further evidence of this split is found by examining usability test guidelines and recommendations from popular usability handbooks, such as Barnum (2002), Hackos and Redish (1998), Nielsen (1993), and Rubin (1994), which confirm the usual approach of testing novices and experts for different reasons. In the case of determining whether a product is sufficiently simple and intuitive for beginner use, or its level of learnability, testing novice users is essential; however, data from all levels of users would be needed to gain a picture of its full range of usability.

These time-honored approaches continue to render valuable usability information. As the field matures, however, it is appropriate for its methods to evolve in sophistication and accuracy. Hackos and Redish (1998) do provide a deeper level of granularity to the field by identifying four levels of users, namely, 'Novices', 'Advanced Beginners', 'Competent Performers', and 'Expert Performers', with sensitive, detailed descriptions of each. This provides more highly developed representations of user populations; however, in their descriptions, they continue to maintain the disparate nature of information that can be gained from each. In another extension of usability approaches, researchers from the Hong Kong University of Science and Technology tested novice users against experienced users, having created the 'experienced' users by providing them pre-test training on application tested. The research design and comparative results suggest that testing and analyzing the performance of novice users against experienced users in the same test provides an additional layer of information that testing the two separately does not provide (Goonetilleke et al., 2001).

To further extend the depth and maturity of usability approaches, the authors sought to develop and perform identical usability tests with users of different experience levels, from novice to expert. Such a method allows for comparison between user levels revealing an additional layer of information relevant to the identification of usability problems, as evidenced in the work discussed below.

The potentially critical importance of such information was highlighted by an occurrence in the medical field as reported by Leveson and Turner (1993), in a discussion of Therac-25, a nuclear accelerator designed to administer radiation treatments to cancer

patients. Over the course of two years, six people were killed or seriously injured upon receiving massive overdoses of radiation during treatment with Therac-25. As was later discovered, one of the things that went wrong was the ability for the expert operators to enter correct prescription information more rapidly than the software could detect and provide feedback on input errors. This was discovered only when a hospital physicist pieced together a sequence of expert-user actions, mastered the sequence and was able to duplicate an accident. These flaws might have been found earlier had novice and expert users been given the same test. Differences in application manipulation would have been highlighted by the marked differences in performance by novice and expert users. The fact that the problem occurred with experts, but not with novices, would have provided early indication of the type of usability problem that existed, namely the inability of the software to respond as rapidly as expert users were able to input information. This suggests that the defect could have been corrected prior to the release for use with live patients.

Dillon and Song (1997) acknowledged this need for a comparison between user levels and conducted a study comparing textual and graphical interfaces for an art-resource database. In a study comparing textual and graphical interfaces for an art-resource database, Dillon administered the same test to two levels of users. The results from the two groups were compared and it was found that expert performance was unchanged with the addition of graphical support, but novice performance improved.

In a further methodological refinement, Uehling and Wolf (1995) describe a prototype of a usability testing tool comparing novice completion time to expert completion time on the same task. In this usability test, novice user action for a specific task was compared to a prerecorded expert user action. The program then returned graphs of the two users that illustrated where there were problems by the display of large differences between user completion times. Even though this advanced usability group performed this type of analysis a decade ago, the field at large has given it little notice and has not moved in that more refined methodological direction.

Kurosu's 'Novice Expert Ratio Method' (NEM) further demonstrates the usefulness of quantitative between-group comparison (Kurosu et al., 2002). Kurosu named the technique and used the NEM, like the Uehling and Wolf usability testing tool, to find novice/expert completion time differences. The large gaps between the two completion times were believed to indicate where usability problems exist. This useful comparison technique taps into the differences between the two users brought about by their relative skill levels.

As these studies show, usability testing typically confines itself to two levels of users: novice and expert. A designated level can reflect either the user's aptitude at general computer usage or the user's familiarity with a specific application. For optimal usability testing results, both general computer aptitude and application familiarity should be taken into account when selecting user participants. The level of each user should be determined by a combination of the two variables. By employing this user-level distinction and testing each one in identical situations with quantitative measures, a comparison of results between the levels is possible. These findings can indicate where there are problems and the types of problems. The study carried out by the authors supports this hypothesis, as discussed below.

2. Method

This study was a structured usability test on a web-based employee time sheet application in use at a 600-employee work facility. The study was performed as an empirical investigation in usability methods; as such, it employed a large sample of participants, in this case, 60. The product studied was a real-world application, allowing the study data to also be employed in usability improvements.

In the timesheet application, users were required to enter daily work hours, absent hours, as well as project and task allocations. The purpose of the original design of the timesheet application was to combine the standard timesheet, used for payroll purposes, with work allocation, used by management for budgeting and reporting purposes (see Fig. 1). User-entered data was required to conform to a number of rules, reflecting the policies and procedures of the employer, and the parameters of each employee's job assignment. Accordingly, the primary window alone contained many possible user operations, with the added complexity layer that some of the operations were required for each real-world use session, while many other operations were required depending on multiple conditions that had occurred in the work environment for any given week.

For purposes of the experiment, the 60 user-participants were each given the same, multi-layered task to perform. The task was presented to each user in the same narrative format, as if that individual were a particular employee, who had worked a certain number of hours that week, had been absent a certain number of hours, had performed specified tasks on various projects, and, then, had to correctly submit the information to management using only the application.

The participants were sampled from three levels of 20 participants each, with the user-level designation depending on two variables: general computer aptitude and experience with the application being tested. The three levels were: (1) novice/novice, inexperienced computer users, less than 1 year of computer experience, who had never used the application being tested; (2) expert/novice, experienced computer users, more than 1 year experience using computers, who had never used the application being tested; and (3) expert/expert, experienced computer users who were experienced with the application being tested. After a year of using this tool one or more times per week, the user's ability and skill was rated at the expert level.

In order to control for variation in computer performance and external environmental factors, all tests were conducted at one desk, on the same computer, with identical physical and software setup and starting points, even down to the starting position of the mouse on the desk, and non-varying lighting conditions. All users were provided with identical tasks and instructions. The real-world facility in which the testing was required to occur was a secure installation in which video-taping was not allowed. This limitation was mitigated by using a test and data collection approach that allowed for rapid, real-time logging of observed behaviors against a standard that was repeatable across all test sessions, and having the same observer perform all test sessions.

The quantitative measures in this experiment were completion time and a measure comparable to user errors, termed in this study as 'user deviations.'. These approaches

Hours Worked

Week Beginning: 16-Oct-98 Total Hours Worked: 40.0 Status: Not Submitted
 Total Reported: 40.0 Task Allocation Reported: 40.0 Get Task Allocation...

	Fri(pm) 10/16	Sat 10/17	Sun 10/18	Mon 10/19	Tue 10/20	Wed 10/21	Thu 10/22	Fri(am) 10/23
Hours Worked:	4	0.0	0.0	8	8	8	8	4
Absent Hours:								

Remarks:

Administrative Comments:

Submit Save Close Reset Help...
 Save to File ***NOTE: Prints ONLY the data that has been SAVED.

Contact wPO6 Administrator at: cmto@arlut.utexas.edu

Weekly Task Allocation

Week Beginning: 10/18/98 Hours Worked: 40.0
 Allocation Unit: hour percentage Total Allocated: 40.0 hours

	USAF tes none	USMC tes none	USN test none
Administrative Support	-	-	-
Code/Debug/Unit Test	20.0	-	20.0
Configuration Management	-	-	5.0
Design	-	10.0	10.0
Document Editing	-	-	-
Installation/Field Support	-	-	-
Meetings: External	-	-	5.0
Meetings: Internal	-	-	-
Other	-	-	-
Personnel Management	-	-	-
Professional Development: External	-	-	-
Professional Development: Internal	-	-	-
Project Planning	-	-	-
Project Tracking	-	-	-
Requirements Analysis	-	-	-
Requirements Management	-	-	-
Security Administration	-	-	-
...	20.0	10.0	10.0

Remarks... Reset Close Help...

Fig. 1. Screenshots of test application main data entry screens.

generated identical types of quantitative data from each test session to allow for between-subject and between-group analyses. There were 45 elements on the usability test, each of which was given an alphabetic identification, A, B, C, D, etc. through SS. An ‘element’ refers to a single part of the tested product on which the user would perform an action to

complete an assigned task. During test preparation, an ideal action on each element was identified, with the set of ideal actions creating an ‘optimal path’ to task completion. For data collection purposes, this set of ideal elements provided a baseline from which to observe the actual user behavior. This is similar to Wayne Gray’s ‘goal structure analysis of errors’ in which user errors are identified in relation to a pre-defined model (Gray, 2000). It was important to recognize, however, that not all user actions that differed from the identified course of action could be accurately called ‘errors.’ For example, a user work-around deviates from the optimal path but since it can still allow for successful completion of the task it is not, in the strictest sense, an ‘error.’ Likewise, there are often multiple paths to a single goal, although that was not the case in this particular study. The data collection process made allowance for this in several ways. The user action is noted in the ‘workaround’ column, with room for a textual note regarding the action actually performed and if it was successful. Noting alternate paths as workarounds in the data collection process, by marking them as a ‘deviations’ from the expected path, can provide usability data that is as valuable as noting using errors. Likewise, a user hesitation might not be captured as an ‘error,’ but if captured as a deviation can highlight a usability problem in the element of the product in use.

In this study, if the user performed the optimal action on an element, no deviation was noted; if the user performed an action other than the optimal one, it was noted as a deviation. For example, the first four elements, or optimal user actions, on the test were: (A) double-click on application icon, (B) wait, (C) click once on login button, (D) type user name. Note that an element could also consist of a system behavior requiring the user to perform a passive behavior as the ideal. In Element C of the sample above, the user was on the initial login screen. The ideal user action in that case was marked as ‘click once on Login’; if the user clicked the wrong button or clicked multiple times, that would be noted as a deviation from the ideal.

The method used to measure usability was as follows: (1) measure the user deviations from the ‘optimal path’, (2) use the user deviations as indicators to potential areas where usability problems exist, and (3) conduct Tukey honestly significant difference (HSD) post-test to infer the type of usability problem. The data collection approach of noting user behaviors on each detailed element of the tested product allowed for user performance on each element to be examined independently across users and test sessions.

User deviations on a specific element indicated that one or more usability problems could exist with that element. For example, Element B was a step in the test where the users were required to wait more than 10 s for the application to load after double-clicking on its icon; the application gave no feedback to the user to indicate the system was working. During this wait period, the users made multiple extra clicks on the screen and icon, hesitated, questioned the screen, and questioned the tester. Each of these actions was captured in the tabular data sheets. Following a univariate analysis on that specific element, with the dependent variable being the number of deviations committed and the fixed factor being experience level, the authors performed a Tukey HSD post-test to identify which of the groups differed significantly from one or both of the others. Note that Tukey’s HSD is a conservative, pairwise comparison between means when the n ’s for each group to be compared are equal.

Table 1

Group means, standard deviations for user deviations logged and time to complete task (in minutes) (Faulkner, 2003)

User deviations by experience level	Mean	<i>N</i>	Std. deviation
Novice/Novice	65.60	20	14.78
Expert/Novice	43.70	20	14.16
Expert/Expert	19.20	20	6.43
Time by experience level	Mean (min)	<i>N</i>	Std. deviation
Novice/Novice	18.15	20	4.46
Expert/Novice	10.30	20	2.74
Expert/Expert	7.00	20	1.86

3. Results

The number of deviations on each element varied from 2 to 280. As expected, on average novice/novice participants committed more deviations than expert/expert participants on each element, and expert/novice results were between the other two levels (see Table 1). Because the three groups represented experience levels with computers in general, and with the tested software, analyzing the specific test elements in light of differences between the three groups by means of a Tukey HSD post-test, as shown in Table 2, provided a statistical clue to the types of usability problems encountered by the users.²

While the initial univariate analysis of the data indicated that the groups differed significantly for user deviations on Element B, the pairwise post-test shows that the novice/novice and expert/novice groups did not differ, but each of those differed from the expert/expert group.

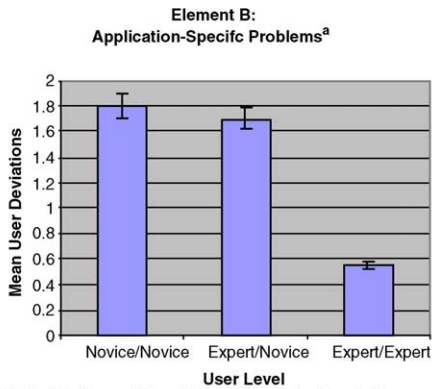
A separate post-test contrast on each of the other elements rendered one of the following types of similarity/difference combinations between the groups:

1. Novice/novice and expert/novice did not differ significantly from each other, but each differed from expert/expert on four of the elements tested (see Element B in Appendix A and Fig. 2).

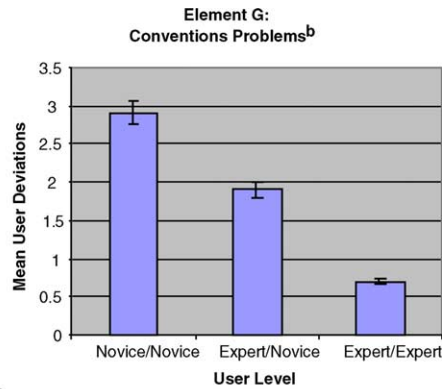
² In comparing users' results based on experience with computers and experience with the test application, there are two dimensions upon which users can differ from one another. When comparing novice/novice or expert/expert to expert/novice, the users only vary on one dimension. When comparing novice/novice to expert/expert, the users vary on two dimensions. In order to determine if the difference between comparing results that vary on one dimension to results that vary on two dimensions would be a confounding factor in the analysis, results from multiple elements were analyzed to see if there was an effect. In some cases the difference between comparing changes of one dimension to changes of two dimensions had a notable difference in one direction, for example novice/novice and expert/expert differed from one another but not from expert/novice (see Element C in Appendix A). While in other cases two-dimensional change only had an effect if it was on the dimension pertaining to the skill level with the test application (see Element G in Appendix A). There was not an effect of a two-dimensional change, but there was an effect in the one dimensional changes. With this shift in effect on comparing a change of one dimension to a change of two dimensions, a confound was not found.

Table 2
Post hoc analysis example: user deviations on element B, a ‘wait for loading’ element

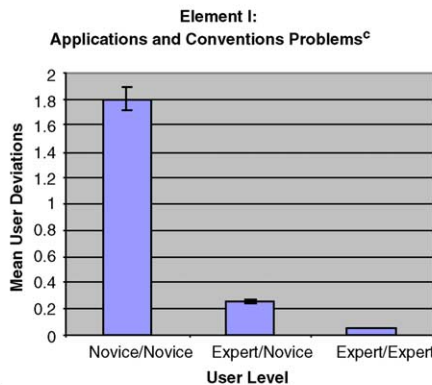
Experience level		Mean difference for user deviations	Significance
Novice/Novice	Expert/Novice	0.10	0.952
	Expert/Expert	1.25	0.001
Expert/Novice	Novice/Novice	−0.10	0.952
	Expert/Expert	1.15	0.003
Expert/Expert	Novice/Novice	−1.25	0.001
	Expert/Novice	1.15	0.003



^aNovice/Novice and Expert/Novice differ significantly from Expert/Novice



^bExpert/Novice and Expert/Expert differ significantly from Novice/Novice



^cAll levels differ from each other

Fig. 2. Examples of usability problems revealed by deviations (dependent variable Tukey HSD groups differed significantly for number of errors committed: Element B, at $F(2,57)=9.427, p < 0.001$; Element I, at $F(2,57)=7.312, p < 0.001$; Element G, at $F(2,57)=18.711, p < 0.001$).

2. There were 10 elements for which novice/novice differed significantly from both expert/novice and expert/expert, but the latter two did not differ from each other (see Element G in Appendix A and Fig. 2).
3. Each level differed significantly from both other levels for two elements (see Element I in Appendix A and Fig. 2).
4. On 10 elements, novice/novice and expert/expert differed significantly from each other, but neither differed from expert/novice (see Element C in Appendix A).
5. Not significant: The means of the groups did not differ significantly on 15 of the elements (see Element A in Appendix A).

Following numerical data collection and statistical analyses, the researchers took the knowledge gained from usability testing observations, data collection and analyses, and based on usability expertise, performed a detailed heuristic analysis of each problem identified. Problems included such things as: an ‘information only’ field that appeared as an entry field and on which users attempted to click and make entries; buttons that lead to required user actions, but which were in a different order than the order in which actions needed to be performed; and a critical data entry field that required the user to tab out of it in order to save the last entry. The latter problem was one of the most significant, occurred equally among novices and experts, and carried the severe consequence of a lost data element, with no warning to the user. Finally, each usability problem noted was paired with one or more criteria from three sets of heuristics, derived from Nielsen and Molich (1990), Kurosu (1997, 1998), and Faulkner (2001). See Appendix B for a sample of the detailed problem description table.

4. Discussion

Relevant insight can be gained from testing multiple levels of expertise on identical, quantitative tests, and comparing between groups. Types of design problems may be revealed by analyzing the data across three levels of users and applying a simple statistical test. The results of this test returned five different types of similarity/difference combinations, which can be explained in the following ways:

1. The elements in which novice/novice and expert/novice did not differ significantly from each other, but both differed from expert/expert suggest there was a problem with the presentation or behavior of those elements that was not attributable to a lack of knowledge about how to use a computer (see Element B in Fig. 2). Those elements failed to provide intuitive or familiar software conventions, but the behavior of that portion of the software could be learned with practice. For example, Element B had four problems with the main problems being that there was a long loading time without the appropriate feedback (see Appendix B).
2. The elements in which novice/novice differed significantly from both expert/novice and expert/expert were easier for experienced computer users to understand, regardless of their previous exposure to this particular piece of software (see Element G in Fig. 2). These elements failed to provide support for the novice user.

3. The elements that differed significantly from one another on all levels indicated a combination of factors related to software conventions and novice-user support (see Element I in Fig. 2).
4. The elements where the novice/novice and expert/expert differed significantly from each other, but neither differed from novice/expert suggest that general knowledge about computers and software makes these elements easier to understand and use, but is not sufficient to entirely prevent user error (see Element C in Appendix A).
5. There are two explanations for the cases in which no significant differences among users was returned (see Element A in Appendix A). In these cases all users had similar experiences with the software regardless of experience level. Where low numbers of errors were committed, usability problems were either not present, or were so minor as to have little effect during testing. Where high numbers of user deviations occurred, the design violated basic usability principles.³

Visual representations of these contrasts and the types of errors they reflect can be seen in Fig. 2, where graphs of errors-by-level on elements B, G, and I are shown as examples.

For example, Element B was a particularly long waiting period, with little visible feedback for the user. Accepted usability guidelines indicate that long processing times should be avoided, and if a wait time is more than 5 s the system should provide a visual indication to wait. Element U appeared to be an entry field, but was not, and required an unusual and inconsistent user action to make it work. Use of element AA required the odd combination of first a mouse click in a certain place, then hitting the Enter key, then removing a placeholder. Element SS was virtually invisible and required special knowledge on the part of the user. Describing these and the other problems in the software gives clues about the types of changes that would support usability.

The field of usability lacks this empirical connection between the existence of a problem and the ability to identify the nature of the problem. Currently, usability tests are able to reveal if a problem exists, possibly even find where it is, but are unable to define the problem or tell what is causing the problem. Naming the problem requires the usability analyst to make an interpretive decision based on experience rather than direct empirical support. Testing multiple users of different levels can reveal more usability problems, of more types, and give an idea of the extent or severity of the most common problems. Deviations by all levels of users can indicate severe and central usability problems, while comparison between user levels reveals a wider variation of problems along with implications regarding where and why they are occurring.

³ Three of the contrasts were significant for only one of the elements and may have been anomalies. Novice/Novice and Expert/Novice differed from each other, but neither differed from Expert/Expert; Expert/Novice and Expert/Expert differed, but neither differed from Novice/Novice; Overall: The means of the groups differed overall, but no single contrast was significant.

5. Conclusion

Current methods require usability practitioners to make cognitive interpretations to connect the existence of usability problems to their identification. Approaching each element as a separate test and contrasting results by user type can provide insight to the type of problem prior to cognitive interpretation by the practitioner. These could be applied and developed by the practitioner in various ways. In most situations, usability practitioners would be applying these comparisons using however many subjects are possible to test within available time and resources, rather than the large number used in this research situation; for ease of analysis, the authors do recommend pre-screening for expertise and categorizing into equal numbers of participants per group. The same ‘number of users’ decision process would apply as in typical usability testing, with the understanding that increasing the number of users tested increases the reliability of the results (see [Faulkner, 2003](#)). While in experience-level situations, that variable could be measured on a continuum and analyzed via correlation, these three distinct levels were selected for this study and analysis for two reasons: (1) the categories fell out naturally from the types of users in this environment and of the application studied; and (2) the distinct levels made for ease of comparison using a simple, post hoc statistical test which was easy to interpret and, perhaps, unique in the insights it could provide. ‘Expertise’ should be defined prior to testing according to the type or types of experience that would be important to a particular application. For example, in an application created for a specialized profession, say, hotel reservation clerks, various categories of expertise could be identified in terms of computer experience (as in this study), hotel reservations experience, other reservations experience, and experience with the application being tested.

The type of information generated by such approaches can benefit usability practitioners in at least two ways. For individuals new to the practice of usability, such information can provide cues to the practitioner for determining the type of usability problem that is occurring when all that can actually be observed are the user behaviors that indicate that a problem exists. It is expected, though, that individuals attracted to the field particularly the most senior practitioners will be skilled at making the necessary cognitive leaps between user behavior the type of usability problem that exists. For senior practitioners, this method is suggested not as a substitute for their specialized skills and experience, but rather as evidentiary support that can be provided to convince product designers, engineers, and decision-makers ([Byrne and Gray, 2003](#)), who might otherwise give less credence to the usability professional’s expert recommendations as ‘opinions’ rather than critical fixes that should be made. Increasing methodological rigor in the usability field has the potential to increase the stature and real-world impact of its special expertise beyond what usability professionals themselves know to be of value in product design, implementation, and fielding.

Appendix A

[Tables A1 and A2.](#)

Table A1

Statistical contrasts of group level by test element (elements A through M and others reported)

Element ID	Total user deviations	Nov/Nov deviations	Exp/Nov deviations	Exp/Exp deviations	Usability violation indicated (see legend)	Element description
A	19	13	4	2	8	db-click app. icon
B	81	36	34	11	1	wait
C	41	24	15	2	4	Single-click 'Login'
D	6	4	2	0	8	Type user name
E	39	21	17	1	8	Activate password
F	2	1	0	1	8	Type password
G	42	36	5	1	2	Single-click 'Login'
H	21	10	5	6	8	Wait
I	110	58	38	14	3	Single-click 'submit timesheet'
J	57	37	6	14	2	wait
K	91	44	33	14	4	Activate 'hours: fri'
L	43	20	18	5	4	Type 4
M	39	24	9	6	2	Activate 'hours: mon'
U	202	19	95	28	1	Activate 'enter absent hours: fri'
AA	280	123	106	51	1	Activate 'code'
BB	48	33	5	10	5	Type 20
CC	128	69	38	21	2	Activate 'testing validation'
KK	52	24	17	11	8	Activate 'project plan'
LL	25	14	9	2	4	Type 5
MM	31	13	15	3	6	(move out of project plan)
SS	120	69	42	9	1	(close applet)

Table A2

Usability violation legend

Identification number	Tukey HSD results (contrasts significant at 0.05)	Usability violation
1	Novice/Novice and Expert/Novice differed from Expert/Expert	Software conventions
2	Expert/Novice and Expert/Expert differed from Novice/Novice	Support for Novice
3	All levels differed from each other	Software conventions and support for novice
4	Novice/Novice and Expert/Expert differed, but neither differed from Expert/Novice	Prevent user deviations
5	No significant differences	Unspecified
6	No significant differences	Unspecified
7	No significant differences	Unspecified
8	No significant differences	Basic usability

Appendix B

Table B1.

Table B1
Detailed analysis of usability problems in test interface: element B

Element ID	Test element	Usability violation	Problem ID	Usability problem description	Nielsen's category	Kurosu's category	Faulkner's comments, categories
B	Wait	Software conventions	3	While the program is loading the user is presented with a large, mostly blank window. The only feedback is a mostly static phrase in a simple font in the bottom left of the window. If response time may be greater than 2 s, there should be an indicator that the system is working, or greater than 10 s, an indicator of estimated time to completion or completion of sequential steps	User feedback	Prompt feedback	
			4	The only feedback that the user receives during the long loading phase are the words 'starting applet'. 'Applet' is a term specific to the programming language and its meaningless to most users	Speak the user's language	Familiarity, cognition	
			5	The 'starting applet' feedback appears in the bottom, left corner of the large, white window, not in a readily visible location	None	Fitness to the body, visibility	Visual design: where
			6	The loading of this program can take between 15 and 45 s. One source suggests: 'Software responses to user actions shall occur within 0.1 s 80%, within 0.5 s 90%, and within 1 s 99% of the time'	User feedback (partial)	Prompt feedback, efficiency	Minimize response time

References

- Barnum, C.M., 2002. Usability testing and research. Pearson Education, Inc, New York.
- Benson, C., 2001, April 6. User testing: and how to strengthen GNOME by doing it. Retrieved March 8, 2004, from http://developer.gnome.org/projects/gup/ut1_pres/img0.htm
- Byrne, M.D., Gray, W.D., 2003. Returning human factors to an engineering discipline: expanding the science base through a new generation of quantitative methods—Preface to the special section. *Human Factors* 45 (1), 1–4 (Spring).
- Dillon, A., Song, M., 1997. An empirical comparison of the usability for novice and expert searchers of a textual and a graphic interface to an art-resource database. *Journal of Digital Information* 1 (1).
- Faulkner, L.L., 2001. Measuring software usability: developing a heuristic evaluation process and tool, Proceedings of the International Conference on Practical Software Quality Techniques, Orlando, Florida 2001.
- Faulkner, L., 2002. Reducing variability—research into structured approaches to usability testing and evaluation, Proceedings of the Usability Professionals Association, Orlando, July 8–12 2002.
- Faulkner, L., 2003. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments & Computers* 35 (3), 379–383.
- Fu, L., Salvendy, G., Turley, L., 2002. Effectiveness of user testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology* 21 (2), 137–143.
- Goonetilleke, R.S., Shih, H.M., Kurniawan, S.H., On, Fritsch, 2001. Effects of training and representational characteristics in icon design. *International Journal of Human–Computer Studies* 55 (5), 741–760.
- Gray, W.D., 2000. The nature and processing of errors in interactive behavior. *Cognitive Science* 24 (2), 205–248.
- Hackos, J.T., Redish, J.C., 1998. User and task analysis for interface design. Wiley, New York.
- Kurosu, M., 1997. Structured heuristic evaluation (sHEM), Conference Presentation, IEEE International Conference on Systems, Man, and Cybernetics 1997.
- Kurosu, M., Sugizaki, M., Matsuura, S., 1998. Structured heuristic evaluation method (sHEM), Proceedings, Usability Professionals Association Annual Meeting, 3–5 1998.
- Kurosu, M., Urokohara, H., Sato, D., Nishimura, T., Yamada, F., 2002. A new quantitative measure for usability testing: NEM (novice expert ratio method), Poster Session Presented at the Annual Conference of the Usability Professionals' Association on Humanizing Design, Orlando, Florida 2002.
- Leveson, N.G., Turner, C.S., 1993. An investigation of the Therac-25 accidents. *IEEE Computer* 26, 18–41.
- Nielsen, J., 1993. Usability engineering. AP Professional, Chestnut Hill, MA.
- Nielsen, J., 2000, February 6. Novice vs. expert users. Jakob Nielsen's Alertbox. Retrieved March 1, 2004, from <http://www.useit.com/alertbox/20000206.html>.
- Nielsen, J., Molich, R., 1990. Heuristic evaluation of user interfaces. In Proceedings of ACM CHI'90 Conference, Seattle, WA, 1–5 April, pp. 249–256.
- Rubin, J., 1994. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. Wiley, New York.
- Straub, K., 2003, September. Pitting usability testing against heuristic review. UI Design Update Newsletter. Retrieved March 8, 2004, from <http://www.humanfactors.com/downloads/sep03.asp>.
- Uehling, D.L., Wolf, K., 1995. User action graphing effort (UsAGE). Paper Presented at the Conference on Human Factors in Computing Systems, Denver, CO.