User Testing in Industry:
A Case Study of Laboratory, Workshop, and Field Tests

Morten Hertzum
Centre for Human-Machine Interaction
Risø National Laboratory, Denmark
morten.hertzum@risoe.dk

Abstract. Applied user testing involves more usability evaluation methods than laboratory tests and is critically dependent upon a number of issues seldom treated in the literature. The development of the system described in this longitudinal, diary-based study evolved around five user tests: a laboratory test, a workshop test, and three field tests. The user tests had a substantial impact on the focus of the entire development effort in that 25% of the primary developer's time was spent solving problems encountered during the tests. The laboratory test made use of set tasks and was biased toward *how* tasks were performed with the system, at the expense of *what* tasks could be performed. The workshop test was more informal and apparently led the users to adopt a more exploratory attitude. Careful arousal and management of the users' commitment to participate actively proved essential to effective user testing, especially during the field tests.

1 Introduction

Slightly caricatured the literature depicts user testing as videotaped usability laboratory tests with set tasks and no context. However, accounts of how user testing is done in practice evidence that applied user testing takes many forms to meet real-life needs and limitations (see, e.g., Brooks, 1994; Szczur, 1994; Zirkler & Ballman, 1994). This study provides field data on the user testing done in a project concerning the development of a graphical front end for an existing application. The purpose is to investigate the effectiveness of certain, very different user testing methods with respect to how well they fit into the development process and how well the tests resemble the use context that the evaluated system is intended to support.

The data collected for this study are a diary that covers the activities of the primary systems developer and the reports from the five user tests the project went through on its way from early prototype to release of the system: a laboratory test, a workshop test, and three field tests. The laboratory test was a conventional thinking-aloud study, except that the evaluators pinpointed misconceptions and other problems on the fly rather than by analysing videotapes. Unlike the laboratory test the remaining user tests emphasised low cost and an informal atmosphere. The workshop test, conducted by the developers in a conference room, consisted in having a group of users work two by two without being closely observed. This bears some resemblance to co-operative evaluation (Wright & Monk, 1991), where designers serve as evaluators of their own systems, and to constructive interaction, a variation of the thinking-aloud study where two users jointly discover how to use a system by trying it out (O'Malley et al., 1984). The field tests were similar to beta tests, hence they were performed by the users without supervision. Smilowitz et al. (1994) find that beta tests may be a cost-effective usability evaluation method.

The effectiveness of a user test depends on a number of interrelated issues. This study compares and contrasts the investigated user testing methods along four dimensions of test effectiveness. This four-dimensional framework is outlined in the next section. Section 3 through 5 describe the studied project, the data collection, and the user tests. The sixth section discusses the effectiveness of the tests; and the seventh section summarises the lessons learned.

2 The Effectiveness of User Testing

Several studies indicate that user testing is currently more effective than other competing approaches to the development of systems that meet the users' needs (Brooks, 1994; Pejtersen & Rasmussen, 1997). This does however not mean that any kind of user testing fits any situation equally well. User testing methods can for example be divided into types according to the approximate point in the development process at which the issues addressed by the test match the major design concerns (Rubin, 1994) or according to the aspect of the use situation at which the test is focused (Pejtersen & Rasmussen, 1997).

The high-level framework applied in this study (see Figure 1) reflects that effective user testing is dependent upon a good fit between the test and the rest of the development process as well as between the test and the use context. The framework identifies four key dimensions of test effectiveness and offers a simple way of illustrating how a user testing method balances these dimensions. Applied user testing is subject to a number of trade-offs, for example between consumed resources and obtained benefits in terms of impact, robustness, and ecological validity as well as between ecological validity and the control required to achieve robustness.

2.1 Ecological Validity

The situations in which user tests are performed differ from the real world in that some aspects of the real world have been left out of the test situation while other aspects that do not exist in the real world may have been added. The closer the test situation is to the real world, the more ecological the test. Ecological gaps between the test situation and the real world introduce a risk that what appears as a problem during a test will not be a problem during actual, real-world use and that some of the problems that will surface during actual use will not surface during a test. While a number of studies compare various usability evaluation methods with laboratory tests, hardly any studies compare evaluations with actual, real-world use of the systems. Thus little is known about to what extent the problems detected during tests are ecologically valid. In a notable study Bailey et al. (1992) found that only two of the 29 problems encountered during a heuristic evaluation had an impact on the users' task completion times and subjective preference.

Thomas & Kellogg (1989) identify four areas of ecological gaps in laboratory tests: *User gaps* are caused by individual differences between users and by a gap between the users' motivation to perform in the laboratory and in their day-to-day work. *Task gaps* are caused by difficulties in generalising from the tasks that can be observed in the laboratory to all the tasks the users

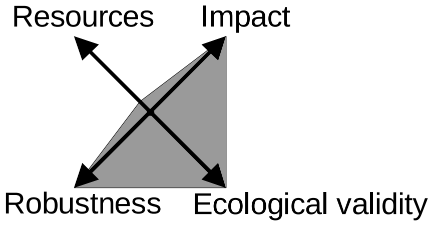

Figure 1. The effectiveness of user testing. The shaded area in the figure indicates the ideal user test that requires a minimum of resources and yields full impact, robustness, and ecological validity.

will want to carry out with the system as well as by differences between a short list of well-defined tasks to be performed in the laboratory and the real world's ongoing stream of possibly ill-defined tasks. *Artefact gaps* are caused by differences between using a single system in the laboratory and using a larger computing environment in the real world and by differences between short-term use in the laboratory and long-term use in the real world. *Work-context gaps* are caused by differences in job context, social context, and cultural context. The way to get around these gaps is to go to the field (e.g., Whiteside et al., 1988), but doing so reduces the robustness of the test.

2.2 Robustness

A method is robust when it produces fairly stable results across a range of minor variations in the test situation. This means that a rerun of the test will yield essentially the same results. The robustness of a user test depends on a number of issues that are within the evaluators' control, such as the number of participating users and the level of detail at which it is prescribed what goes on during the test sessions. However, user tests are also affected by a number of issues that are beyond the evaluators' control and thus vary from one instance of the test to the next. One such issue is the users' reaction to the often stressful test situation. A typical laboratory test involves trying to use a new system, being videotaped, and performing in front of others. Each of these three circumstances is experienced as unpleasant by many people and their combination creates a situation that is stressful to most people (Schrier, 1992).

A user testing method must be both robust and ecologically valid to reliably predict the parts of a system that need to be changed because they are confusing, slow users down, or do not match the users' needs. Whereas it is a primary intention of laboratory tests to provide a controlled environment where various sources of variability can be kept to a minimum, the resulting robustness is achieved at the expense of reduced ecological validity. Many dimensions of laboratory tests have been investigated including the sufficient number of users (e.g., Lewis, 1994), individual versus co-operating users (Hackman & Biers, 1992), the level of experimenter intervention (Held & Biers, 1992), and various methodological pitfalls (Holleran, 1991). Recently, Jacobsen et al. (1998) have shown that laboratory tests are subject to a considerable evaluator effect in that different evaluators, who analyse the same test sessions, detect markedly different sets of problems. Field tests refrain from tight control of the test situation and thus achieve their higher ecological validity at the expense of reduced robustness.

2.3 Impact

The impact of a user test is its ability to bring about changes in the evaluated system or the development process. That is, the impact concerns the persuasive power of the test rather than its predictive power, i.e. its ability to predict what aspects of the system that will cause problems to users during real-world use. The impact of a user test is most directly seen in relation to the development team but a user test may also interact with other actors in the development organisation and its outcome may therefore also have an impact on, for example, management or marketing (see Brooks, 1994; Zirkler & Ballman, 1994).

Whenever a problem predictive of actual use is left unaddressed an opportunity to improve the evaluated system is missed and the effort that went into finding the problem is wasted. However, the time required to fix a problem must be weighted against the benefit of fixing it and the benefit of spending the same amount of resources on any other outstanding task. Sawyer et al. (1996) define the impact ratio of a usability evaluation as the number of solved problems divided by the total number of problems found, expressed as a percentage. On the one hand, this way of calculating impact incorrectly assumes that all problems are equally severe and that problems are either left unaddressed or solved completely. On the other hand, the impact ratio is easy to understand and calculate, and it provides a rough measure of the action taken in response to a usability evaluation. Sawyer et al. (1996) report an impact ratio of 78% averaged over ten usability inspections, but their calculations are based on the developers' commitment to fix a certain number of problems so the achieved impact ratio may be lower. Whiteside et al. (1988) tested multiple versions of a single system and report an impact ratio of 65% for the early, in-house tests and 48% for the subsequent field tests.

2.4 Resources

Several studies provide formulas to estimate the costs of conducting user tests and try to justify these costs by converting the estimated benefits of performing the tests into cost savings (see, e.g., Bias & Mayhew, 1994). However, despite logical arguments to the contrary the subjective experience of many developers is that usability work lengthens the projects, adds expenses, and fails to prevent that new problems show up when the systems are released for actual use (Lund, 1997). As a result practitioners tend to be cautious and show a strong preference for methods that are low-cost in terms of the time, expertise, and equipment required to apply them. For example, Rowley (1994) reports how a mobile usability testing facility can provide a low-cost alternative to a dedicated usability laboratory. Nielsen (1993) advocates that though low-cost, discount evaluations are inferior to expensive, deluxe evaluations, discount evaluations are highly cost-effective and vastly superior to doing no evaluation work at all.

The choice of user testing method directly affects the cost of finding problems in a system, but the total cost of user testing also includes the cost of addressing the detected problems. This cost may be affected by how well the test fits into the development process but apart from that the cost of addressing the problems is not affected by the choice of user testing method. If the total cost of user testing is dominated by the cost of addressing the problems, the amount of resources that goes into the conduct of a test becomes less critical. Unfortunately, little is known about how the cost of conducting user tests compares with that of addressing the detected problems.

3 The F&N Project

The company where the case study took place is a publicly owned software house with around 2100 employees. The investigated project, initiated in December 1994 and completed in January 1996, consisted in the development of a graphical user interface for the Filing and Notification (F&N) system which contains information about citizens for use by local authorities. Most importantly, the F&N system gives a complete overview of the business that the local authorities have with a citizen in terms of business files which deal with tax, social security, kindergarten, or school. The F&N system was developed in the 1970's as a mainframe application and is used daily by several thousand civil servants in the Danish municipalities. However, efficient use of its character-based user interface requires dedicated training, regular use, and an extensive printed manual.

The purpose of the F&N project was to make a Windows version of the F&N system. To minimise the development time and cost of this version it was decided to implement it as a graphical front end on top of the existing mainframe application. Thus, the information displayed in the front end was to be obtained from the mainframe application, which would run in the background, and data entered into the front end were to be transferred to the mainframe application for updating the central database. This approach preserves the investment in the mainframe application and provides a way to gradually migrate legacy systems to a client/server environment. Since the project concerned the development of a new front end for an existing system, the task consisted chiefly in the design and implementation of a user interface. The amount of analysis was comparatively smaller.

The F&N project group consisted of a project manager, a primary systems developer who was the mainstay of the project, a secondary systems developer who was responsible for the development of a module for handling free-text notes, and an online help writer. The primary systems developer was identical to the present author who was at that time employed in the organisation where the study took place.

4 The Collected Data

The data collected from the project were the reports from the user tests and a diary that covered the activities of the primary systems developer. These data were supplemented with this author's first-hand knowledge of the project.

4.1 Problems

In classifying the problems encountered during the user tests the present study distinguished between utility, defined as ”the question of whether the functionality of the system in principle can do what is needed”, and usability[1], defined as ”the question of how well users can use that functionality” (Nielsen, 1993). The following categories were used to classify the problems: (1) *Utility problems*. A facility or piece of information needed by the users was not present in the system. Example: A citizen's marital status is a prominent piece of information because it determines whether the citizen is entitled to a large number of social security benefits, such as support for single-parent families. However, the field with the citizen's marital status was not really useful because it was not accompanied by the date from which the marital status was valid. (2) *Usability problems*. A facility or piece of information was present in the system but the user remained unaware of it, misinterpreted it, or had trouble using it. Example: When searching for a citizen's social security number on the basis of his or her name a checkbox enabled the user to indicate whether the full name was given or just part of it. The caption of this checkbox, 'Search for patterns', was unintelligible. (3) *Program bugs*. A facility or piece of information was present in the system but due to a program bug it did not behave as intended or did not work at all. Example: The same social security number could appear several times in the drop down list containing the citizens for whom information had most recently been displayed. (4) *Other*. This catch-all category contained only one problem which was related to the system configuration.

To indicate whether or not the problems were solved, each problem was assigned a status: (1) *Solved*, the problem was fixed. (2) *Reduced*, the problem was partly, but not fully, fixed. (3) *Unaddressed*, the problem was either deferred or rejected. A similar assessment of problem status was made during the project to maintain an overview of the progress made. The primary systems developer and either a usability specialist or the project manager made this assessment. The assessment made during the project and the subsequent coding made for this article assigned the same status to 88% of the problems, the remaining 12% were reassigned to indicate that they had been addressed to a lesser extent.

4.2 Activities

The activities of the primary systems developer were tracked in a diary (for a discussion of the diary method see Rieman, 1993). The diary, which was updated successively throughout the day, covered the nine-month period from the first user test through the fifth and contained

[1] Note that the definition of usability used in this study is narrower than the definition in ISO9241-11.

Table 1. The user tests

User test	Number of users	Offset from project start	Test duration	Test conducted by
Laboratory test	6	5 months	2 days	Usability specialists
Workshop test	8	8 months	1 day	Developers
Field test 1	8	10 months	3 weeks	Users
Field test 2	8	12 months	2 weeks	Users
Field test 3	8	13 months	5 weeks	Users

every activity with a duration of 15 minutes or more. The recordings were made on diary sheets, one for each day, and gave the starting and ending time of the activity, the project to which the activity pertained, and a terse description of the activity (see Figure 2 for an example of diary entries). To achieve this level of detail the current diary sheet was lying easily accessible on the developer's desk.

To enable investigations of when a problem was addressed and how long it took to correct it, each activity recorded in the diary was linked to the problems it addressed. Some activities were not performed in response to any of the problems identified by the users; other activities contributed to the solution of several problems. Thus, an activity could be linked to any number of problems, just as a problem could be linked to any number of activities.

5 The User Tests

The F&N project evolved around a series of user tests, each a major project milestone (see Table 1). Though different in many respects the five user tests shared the defining characteristic that a group of target users got hands-on experience with a running system prototype and expressed their opinion about it. The users participating in the tests were regular users of the mainframe version of the F&N system.

5.1 The Laboratory Test

The first user test was conducted by usability specialists in the in-house usability laboratory and involved six users who were asked to think out loud while solving eleven set tasks. Each task consisted of a brief description of a realistic scenario followed by a question, e.g. to find a specific piece of information. The test sessions, one for each user, consisted of a short introduction, 1½ hours of testing, and a debriefing interview. While working with the tasks the users were alone in the test room, and the two usability specialists conducting the test were in the control room which was separated from the test room by a one-way mirror. The users were frequently asked questions such as “What do you think that command will do?”, “What would you expect to see or be able to do at this point?”, and “What do you think the information on that part of the screen is telling you?” The usability specialists recorded problems observed during the sessions and communicated them to the development group in a test report. In addition, the primary systems developer observed the test sessions from an observation room. The sessions were also videotaped but the videotapes were not used during analysis, they were just back-up.

12:50 - 13:10	F&N	Write a help topic for [the online help writer].
13:10 - 14:00	F&N	Add the possibility of opening the window ‘Information about authority’ by double clicking a line in the list of back information about a citizen, including validation of any social security number in that line.
14:00 - 15:10	[Another project]	Assist [a colleague] in the proper use of [a reusable component developed as part of the F&N project].

Figure 2. An excerpt of the diary sheet for August 3, 1995

5.2 *The Workshop Test*

The second user test was also performed in-house but under the management of the development group and in a conference room rather than the usability laboratory. The workshop test began with a guided tour of the F&N system, performed by the project manager and the two systems developers. Then the users had two two-hour sessions for testing, separated by lunch. Finally, the test was concluded by a plenary discussion. Eight users participated in the test and they worked two by two on a number of set tasks, which collectively included five scenario descriptions each followed by a series of about ten specific questions. Each pair of users sat at a separate table with one computer and access to the printer within the room. When the users discovered a problem they either called upon a developer to report it directly or made a print-out of the screen and annotated it. The developers circled among the users to observe, inquire, and receive feedback. When the developers discovered a problem they approached the user for further details. After the test the development group produced an annotated list of the encountered problems.

5.3 *The Field Tests*

The third, fourth, and fifth user test were performed on site and managed by the users themselves. The users, the same eight persons as in the workshop test, had the F&N system installed at their personal workplace and used it occasionally in the execution of their day-to-day duties. They also had opportunity to discuss the system with their colleagues. There was no set tasks to be solved during these tests and the users did not keep a log of how much time they spent testing the system. The developers contacted the users once or twice during a test to ensure that everything was in working order, motivate further testing, and get feedback. Problems discovered by the users were reported by telephone or on the test form to be returned at the end of the test. The development group concluded each test by compiling an annotated list of the reported problems.

5.4 *The Rationale for the User Tests*

The laboratory test and the workshop test were carried out while the F&N system was under formation and receptive to suggestions for both minor and major modifications. The purpose of these two tests was quite similar. A major reason for choosing the workshop test the second time was that it was independent of the busy schedule of the laboratory. The remaining user tests were progressively more concerned with errors, at the expense of inconveniences. The first field test was performed to expose the F&N system to real-life conditions and, thus, have it evaluated in the context of the users' day-to-day duties, workload, and technical environment. The second field test was an informal acceptance test, and the third was the formal acceptance test intended to confirm that the system was ready for release.

6 Discussion

The diary contained 604 hours of work on the F&N project (51% of the working hours) distributed over 129 of the 164 working days covered by the diary. Most of the remaining time was spent on three other projects. The user tests encountered a total of 77 problems, distributed quite unevenly across the tests (see Table 2). One reason for the different number of problems found in each test was that the tests formed a sequence where one test followed another when all or most problems from the previous test had been addressed. Hence, it *cannot* be inferred that the laboratory test was superior to the workshop test, which in turn was superior to the field tests.

6.1 The Ecology of Set Tasks

The user tests uncovered a notably different mix of utility problems, usability problems, and program bugs (see Table 2). Of the problems uncovered by the laboratory test 76% concerned the usability of the system. Probably, a major reason for this was the use of set tasks, which were solved one by one without much digression. Set tasks tend to preclude discussion of whether the system lacks support for some aspects of actual tasks (Wright & Monk, 1991). Further, several users seemed to feel a remarkable pressure to perform during the laboratory test, even though they were told that the object of the test was the system, not the person using it. Under such circumstances the users cannot be expected to notice shortcomings of the set tasks or digress much from them to try other things. The users digressed from the ideal way of solving the tasks—they got into problems and recovered from these problems—but they kept pursuing the tasks and did so with little attention to their ecological validity. Hence, the laboratory test was biased toward usability at the expense of utility. Another consequence of the users' narrow focus on the set tasks was that few program bugs were encountered since the users stayed within the well-tested parts of the system.

The workshop test made use of set tasks too but only 20% of the encountered problems were usability problems. Instead, utility problems and program bugs each made up 40% of the problems. This seems to indicate that the users felt free to go beyond the set tasks and explore additional aspects of the system. In doing so they tested the system against their actual tasks, and they exercised the system in ways not foreseen by the developers. It seems reasonable to ascribe the users' more exploratory attitude to two circumstances:

- Working two by two the users were not alone when they got stuck or in doubt, and differences in their day-to-day work practices fostered discussion and divergent suggestions for solving the tasks.
- The informal atmosphere brought about by the presence of several other users, the face-to-face way of communicating with the persons conducting the test, the one-hour lunch break, and the absence of detailed observation of the users' behaviour.

The workshop test showed that the implications of set tasks were very much dependent upon the test situation in which the tasks appeared. The workshop test also showed that the developers were able to conduct a user test of their own system with results that made the test worth the effort. The workshop test was, however, restricted by leaving it almost entirely to the users to detect the problems, a restriction most obviously addressed by calling in a usability specialist skilled in observing users and spotting their problems.

Collectively the field tests uncovered a broad mix of problems but while the first and third field test were reasonably effective the second failed completely. Field test 3 uncovered a number of utility and usability problems and seemed successful in testing the system against the users' actual tasks. The test uncovered for example several problems relating to the print-

Table 2. Problem classification

Test	Utility problem	Usability problem	Program bug	Other	Total problems found
Laboratory test	8	29	1		38
Workshop test	8	4	8		20
Field test 1	1	2	5		8
Field test 2					0
Field test 3	4	6		1	11
Total	21	41	14	1	77

Note. Since the F&N system evolved from one test to the next the total number of problems found during a test is *not* evidence that one test is better than another.

outs produced by the system. Unintentionally, the set tasks used during the in-house user tests were so focused on the software that the users were not asked to evaluate the print-outs. This highlights a fundamental limitation of set tasks: They make tests blind to aspects not covered by the tasks. The low cost and valuable output of the field tests confer with the findings of Smilowitz et al. (1994), but the F&N project provides no support for their suggestion that the field test method may be improved further by providing set tasks.

6.2 Robustness and the Management of User Commitment

The effectiveness of user tests is critically dependent upon the active participation of the involved users. One way to control this dependency is to prescribe the users' behaviour in detail and carefully observe their execution of these prescriptions, i.e. the approach of the laboratory test. Leaving more decisions and initiative to the users, tests become increasingly dependent upon the individual user's personal motivation to do a good job. Field tests owe their low cost to leaving almost everything to the users, and precisely for that reason the essential task left with the persons conducting field tests is the management of the users' commitment to perform a thorough test. This is an indirect way to strive for robustness, but without supervision and a controlled environment it is practically the only one left.

The laboratory test and the workshop test had scheduled, supervised sessions dedicated to testing, and the detected problems were reported immediately. This way little effort was required on the part of the users to set up these tests and report their outcome. During the field tests it was the users' responsibility to devote some time to testing, and the procedure for reporting problems was more laborious in that it required either describing the problems in writing or phoning one of the developers. Field test 1 was the first time the F&N system was exposed to real-life conditions and it was accompanied by a strong commitment from the development group to fix a substantial fraction of the problems encountered. This context and one or two phone calls from the developers during the test motivated the users to spend some time testing the system. Field test 2 did not have a clearly stated purpose that differentiated it from the other tests. It merely asked the eight users from the two foregoing tests to test the system for the third time, and apparently that did not occasion the necessary enthusiasm. Users participating in a test have a long-term interest in the quality of the system since they will, probably, also be users of the released system, but they need further encouragement. It is the responsibility of the persons conducting a test to convince the users of its importance, otherwise busy users are unlikely to give a test priority at the expense of their day-to-day duties. Field test 3 was unique in that it was performed to decide whether the F&N system could be released or had to go through another iteration. The formal purpose of this test motivated the users since it was probably their last opportunity to affect—this release of—the F&N system.

Since the F&N project concerned the development of a new front end for an existing application the same data were available during the field tests as through the existing mainframe version. Data entered in either version could be viewed with the other. Thus, the users could shift freely between the old and the new version without loss of data or need of rekeying. This complete access to production data was, however, not exploited in the setup of the field tests though it seems that, for example, one-hour sessions where the new version was used in place of the old would have been a cogent tool in the management of the users' commitment. The complete access to years of production data distinguishes the F&N project from the development of new systems from scratch, but it should be noted that vast numbers of projects involve the development of new versions of existing systems. Zirkler & Ballman (1994), who consider field tests necessary to effective user testing, manage user commitment by physically arriving at the users' workplace and conduct the test. Though this is more costly than the field tests of the F&N project, Zirkler & Ballman report significant cost savings over the in-house usability tests they used to run.

6.3 Problem Impact

In the F&N project, 55 problems were solved or reduced while 22 problems were left unaddressed. To take the reduced problems into account the impact ratio calculation in Sawyer et al. (1996) was modified by counting the reduced problems as 50% solved:

$$\text{Impact ratio} = \frac{\text{Solved problems} + 0.5 * \text{Reduced problems}}{\text{Total problems found}} * 100$$

The overall impact ratio was 65%. This is comparable to the 78% reported by Sawyer et al. (1996) and to the 65% and 48% found by Whiteside et al. (1988) for their early in-house tests and subsequent field tests, respectively. However, the impact ratio varied considerably from one user test to another (see Table 3). Looking at the impact ratios of the individual user tests it is striking that only the three first tests had an impact while half of the unaddressed problems were uncovered during field test 3. This reflects that the field tests were increasingly concerned with critical problems only. Problems perceived to be non-critical were more and more often recorded and left for the next version. The 22 unaddressed problems form four groups: (1) Six problems were considered not to be predictive of actual use. Example: It was suggested to add a second confirmatory step after the user had confirmed a new note by pressing 'OK' rather than 'Cancel'. (2) Six problems stemmed from circumstances beyond the developers' influence. Example: The font size, prescribed by a mandatory corporate standard, was considered too small by several users. (3) Five problems were considered to be merely cosmetic. Example: The left margin of the print-outs was slightly narrow. (4) Five problems were left unaddressed because it was considered more important to get the front end released.

A very influential factor in determining whether or not a problem was addressed was *when* the problem was found: Finding a problem early profoundly increased its chances of being addressed. In the beginning and middle of a project much work remains and the project will be one of the project members' major concerns for some time to come. Also, many minor problems can be corrected at almost no extra cost when they can be addressed along with other problems concerning the same part of the design. Near the end of a project most project members spend the majority of their time on other projects or they are about to enter other projects, and little room and will is left for prolonging the old project even moderately. This means that relative to the project members' other responsibilities the time required to solve a user test problem tends to appear reasonable in the beginning or middle of a project and prohibitive near the end of the project (see also Kumar, 1990). Often, the last user test will have a 0% impact ratio because any modifications made trigger an extra test to assess the quality of these modifications.

Table 3. Problem status and impact ratio

Test	Solved	Reduced	Unaddressed	Total problems found	Impact ratio
Laboratory test	25	6	7	38	74%
Workshop test	13	3	4	20	73%
Field test 1	7	1		8	94%
Field test 2				0	-
Field test 3			11	11	0%
Total	45	10	22	77	65%

In the F&N project the single-most important factor in ensuring a high impact of a user test seems to be to avoid performing the test during the last third of the project. The low-cost, unsupervised field tests performed in the F&N project can only be conducted late in the project when the system prototype is fairly stable. This suggests that the price of the low cost of these tests is that they will usually have a low impact. A potentially attractive alternative is supervised field tests where a developer or usability specialist accompanies the system in the field to handle problems with the system and to observe. Supervised field tests can be performed earlier and allow for better management of the users' commitment, but they require more resources and the presence of an evaluator introduces an ecological gap.

6.4 Resources Spent Finding Versus Addressing Problems

The total cost of user testing is the cost of conducting the tests plus the cost of addressing the detected problems. In the F&N project, the laboratory test was the more resource-demanding test in terms of equipment, expertise as well as person-hours, and the field tests required the fewer resources. More notably, the cost of addressing the problems was quite substantial. The primary systems developer spent 25% of his time fixing problems encountered during the five user tests (see Table 4). At the time of the laboratory test several facilities were not yet developed. Thus, the problems found during this test were added to an already long list of outstanding tasks. As the project progressed the list got shorter and increasingly dominated by the input from the user tests. To the primary systems developer this meant that user test issues came to occupy more of his time. Near the end of the project the action taken on the user tests was restricted to the presumably critical problems and the amount of time spent on user test issues dropped. Averaged over the entire project the primary systems developer spent 2 to 3 hours a problem, but the time spent on the individual problems varied a lot. For field test 1 the average time spent to detect a problem was half an hour. Thus the cost of addressing the problems encountered during this test clearly exceeded that of finding them. Similar figures cannot be given for the other tests because field test 1 was the only test administered by the diary-keeping, primary systems developer alone. However, the average time spent to detect a problem was most likely higher for the laboratory test and the workshop test.

During the five months from the laboratory test to field test 1 the primary systems developer spent an average of more than five hours a working day on the F&N project. During the remaining four months of the project it occupied substantially fewer daily hours. A major reason for this decrease in project intensity was the duration of the field tests. To allow the users time to fit the field tests into their schedule and get to actually use the system for some time, the field tests lasted 3, 2, and 5 weeks. During these periods the list of outstanding tasks contained few, if any, high-priority tasks and little work was done on the F&N project. These periods of waiting allowed the project members to devote their attention to other projects, but these periods were also a significant cost of the field tests because they prolonged the F&N

Table 4. Time spent by the primary systems developer, in total and to address the problems encountered during the user tests

Period	Total hours spent	Total hours spent a working day	Hours spent fixing problems	Percent of time spent fixing problems
Laboratory test - workshop test	279	6:29	56	20%
Workshop test - field test 1	216	4:48	70	32%
Field test 1 - field test 2	79	2:09	25	32%
Field test 2 - field test 3	4	0:40	0	0%
Field test 3 - system released	26	0:47	1	4%
Total	604	3:41	152	25%

project. The way to reduce this cost is to make the field tests more efficient, i.e. to obtain the same benefits in a shorter span of time. This is to a certain extent a trade-off between spending resources passively by prolonging the projects and spending resources actively, for example by conducting supervised field tests in order to manage the users' commitment better.

7 Conclusion

This study has investigated the effectiveness of the five user tests conducted during the development of a graphical front end for the F&N system. Overall the user tests were effective and led to numerous improvements of the front end, but the tests differed substantially in terms of how they balanced resources, impact, robustness, and ecological validity. This study concerns the F&N project and no strong claims can be made as to the generality of the findings. The F&N project employed an iterative, user-centred approach and concerned the development of a new version of an existing application. It is reasonable to assume that the findings can be made subject to some generalisation if a development process of a similar nature is studied.

Tests like the laboratory test are designed to control variability and thereby achieve robustness. This is costly in terms of resources such as equipment, expertise, and person-hours, and it introduces a number of threats to the ecological validity of the test. The laboratory test relied on the use of set tasks to direct the users' activities and provide the evaluator with knowledge about what the users were trying to achieve. However, the formality of the laboratory test seemed to place the users under a pressure that precluded considerations about whether the tasks were representative of the users' actual work. That is, the test sessions provided little basis for evaluating the ecological validity of the set tasks. This made the laboratory test less suited early in the development process where the utility of the system, i.e. *what* the system can do, was the major design concern. The laboratory test was more concerned with usability, i.e. *how* tasks are performed with the system.

The workshop test focused more on utility, and it seems probable that this focus was nourished by the more informal test situation with plenty of room for discussion. For that reason it should be considered to swap the workshop test and the laboratory test. It is however worth noting that much of the formality of the laboratory test could be removed at no cost, for example the evaluator could have been in the test room with the user since they talked much anyway. The workshop test combined set tasks and direct supervision with an informal atmosphere and co-operating users. This way the test unfolded around the set tasks with frequent exploration of issues that went beyond the tasks. Since the workshop test left the detection of problems almost entirely to the users, the test did however miss the problems where the users themselves were not aware of their difficulties or ascribed them to their lack of experience in using the system.

The field tests displayed dramatic differences in their contributions to the project. Larger robustness is utterly needed, and since field tests are characterised by leaving almost everything to the users the one, essential task left with the persons conducting unsupervised field tests is the careful arousal and management of the users' commitment to perform a thorough test. Few development tasks could be done in parallel with the field tests since the system had to be rather complete before it could be tested in the field. For that reason, the field tests prolonged the project by introducing week-long periods where the project merely awaited the results of the field tests. These periods of waiting added a substantial cost to the field tests which otherwise required very few resources.

The user tests that were conducted early in the development process had a much higher impact than those conducted near the end of the project where the project members had got seriously involved in other projects. This affected especially the field tests, and it highlights the importance of designing user testing methods in ways that allow them to be applied early. Altogether the user tests had a substantial impact on the focus of the entire development effort in that 25% of the primary systems developer's time was spent solving problems encountered during the tests. This gives an indication of the cost of addressing the encountered problems, and it suggests that the amount of user testing that can be done in a project is limited by the cost of addressing the problems, rather than by the cost of conducting the tests.

8 Acknowledgements

This work has been supported by a grant from the Danish National Research Foundation. I wish to thank Niels Jacobsen and Erik Frøkjær for their valuable comments to earlier drafts of this article. Special thanks are due to the members of the F&N project group, the usability specialists, and the users who participated in the user tests.

9 References

Bailey, R. W., Allan, R. W., & Raiello, P. (1992). Usability testing vs. heuristic evaluation: A head-to-head comparison. In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 409-413). Santa Monica: Human Factors Society.

Bias, R. G., & Mayhew, D. J. (Eds.). (1994). *Cost-justifying usability*. Boston: Academic Press.

Brooks, P. (1994). Adding value to usability testing. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods* (pp. 255-271). New York: John Wiley.

Hackman, G. S., & Biers, D. W. (1992). Team usability testing: Are two heads better than one? In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 1205-1209). Santa Monica: Human Factors Society.

Held, J. E., & Biers, D. W. (1992). Software usability testing: Do evaluator intervention and task structure make any difference? In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 1215-1219). Santa Monica: Human Factors Society.

Holleran, P. A. (1991). A methodological note on pitfalls in usability testing. *Behaviour & Information Technology*, 10, 345-357.

Jacobsen, N. E., Hertzum, M., & John, B. E. (1998). The evaluator effect in usability studies: Problem detection and severity judgments. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 1336-1340). Santa Monica: Human Factors and Ergonomics Society.

Kumar, K. (1990). Post implementation evaluation of computer-based information systems: Current practices. *Communications of the ACM*, 33(2), 203-212.

Lewis, J. R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors*, 36, 368-378.

Lund, A. M. (1997). Another approach to justifying the cost of usability. *ACM Interactions*, 4(3), 48-56.

Nielsen, J. (1993). *Usability engineering*. Boston: Academic Press.

O'Malley, C. E., Draper, S. W., & Riley, M. S. (1984). Constructive interaction: A method for studying human-computer-human interaction. In *Proceedings of the IFIP INTERACT'84*

First International Conference on Human-Computer Interaction (pp. 269-274). Amsterdam: Elsevier.

Pejtersen, A. M., & Rasmussen, J. (1997). Effectiveness testing of complex systems. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (pp. 1514-1542). New York: John Wiley.

Rieman, J. (1993). The diary study: A workplace-oriented research tool to guide laboratory efforts. In *Proceedings of the ACM/IFIP INTERCHI'93 Conference on Human Factors in Computing Systems* (pp. 321-326). New York: ACM Press.

Rowley, D. E. (1994). Usability testing in the field: Bringing the laboratory to the user. In *Proceedings of the ACM CHI'94 Conference on Human Factors in Computing Systems* (pp. 252-257). New York: ACM Press.

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

Sawyer, P., Flanders, A., & Wixon, D. (1996). Making a difference - The impact of inspections. In *Proceedings of the ACM CHI'96 Conference on Human Factors in Computing Systems* (pp. 376-382). New York: ACM Press.

Schrier, J. R. (1992). Reducing stress associated with participating in a usability test. In *Proceedings of the Human Factors Society 36th Annual Meeting* (pp. 1210-1214). Santa Monica: Human Factors Society.

Smilowitz, E. D., Darnell, M. J., & Benson, A. E. (1994). Are we overlooking some usability testing methods? A comparison of lab, beta, and forum tests. *Behaviour & Information Technology*, 13, 183-190.

Szczur, M. (1994). Usability testing—on a budget: A NASA usability test case study. Behaviour & Information Technology, 13, 106-118.

Thomas, J. C., & Kellogg, W. A. (1989). Minimizing ecological gaps in interface design. *IEEE Software*, 6(1), 78-86.

Whiteside, J., Bennett, J., & Holtzblatt, K. (1988). Usability engineering: Our experience and evolution. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (pp. 791-817). Amsterdam: Elsevier.

Wright, P. C., & Monk, A. F. (1991). A cost-effective evaluation method for use by designers. *International Journal of Man-Machine Studies*, 35, 891-912.

Zirkler, D., & Ballman, D. R. (1994). Usability testing in a competitive market: Lessons learned. *Behaviour & Information Technology*, 13, 191-197.