Proceedings of the NordiCHI Workshop on Improving the Interplay Between Usability Evaluation and User Interface Design

Edited by

Kasper Hornbæk
Department of Computer Science
University of Copenhagen

Jan Stage
Department of Computer Science
Aalborg University

HCI Lab Report no. 2004/2

Published December 14, 2004

Improving the Interplay between Usability Evaluation and User Interface Design

Kasper Hornbæk
Department of Computing
University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen
Denmark
kash@diku.dk

Jan Stage
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7, DK-9220 Aalborg
Denmark
jans@cs.aau.dk

Abstract

This paper provides an overview of a full-day workshop that was held on October 23 2004 in connection with the Third Nordic Conference on Human Computer Interaction (Nordichi 2004). The proceedings from the workshop are available from http://www.cs.aau.dk/~jans/events.html.

The ideas and theme of the workshop are motivated and an outline of the contents of the papers that were presented in the workshop is given. In addition we summarize some challenges to the interplay between usability evaluation and user interface design agreed upon at the workshop, as well as some solutions that were debated.

1. Introduction

Software development is highly challenging. Despite many significant successes, several software development projects fail completely or produce software with serious limitations, including (1) lack of usefulness, i.e. the system does not adequately support the core tasks of the user, (2) unsuitable designs of user interactions and interfaces, and (3) lack of productivity gains or even reduced productivity despite heavy investments in information technology (Gould & Lewis 1985, Strassman 1985, Brooks 1987, Matthiasen & Stage 1992, Nielsen 1993, Attewell 1994, Landauer 1995).

Broadly speaking, two approaches have been taken to address these limitations. The first approach is to employ evaluation activities in a software development project in order to determine and improve the usability of the software, i.e. the effectiveness, efficiency and satisfaction with which users achieve their goals (ISO 1998, Frøkjær et al. 2000). To help software developers' work with usability within this approach, more than 15 years of research in HCI has created and compared techniques for evaluating usability (Lewis 1982; Nielsen & Mack 1994).

The second approach is based on the significant advances in techniques and methodologies for user interface design that have been achieved in the last decades. In particular, researchers in user interface design have worked on improving the usefulness of information technology by focusing on a deeper understanding on how to extract and understand user needs. Their results today constitute the areas of participatory design and user-centered design (e.g. Greenbaum & Kyng 1991, Beyer & Holtzblatt 1998, Bødker, Kensing & Simonsen 2004).

However, the interplay between these two approaches, and between the activities they advocate to be undertaken in software development, have been limited. Integrating usability evaluation at relevant points in user interface design with successful and to-the-point results has proved difficult. In addition, research in HCI and software design has been done mainly independently of each other with no in substantial exchange of results and sparse efforts to combine the techniques of the two approaches. Larry Constantine, a prominent software development researcher, and his colleagues express it this way: "Integrating usability into the software development process is not easy or obvious" (Juristo et al. 2001, p. 21).

2. Idea of the Workshop

The idea of the workshop was to inquire in more detail into the interplay between design and usability evaluation. Software development is the overall process that we focus on. Within this process there is a multitude of different activities. Two key activities are user interface design and usability evaluation, see figure 1. The purpose of usability evaluation is to assess the usability of user interface designs. This assessment is based on different design products, e.g. mockups, prototypes, incomplete versions of the final system or even the final system itself. In the usability evaluation activity these design products are assessed and the results are fed back into the user interface design activity. The results can also take a variety of forms, e.g. the traditional usability report with problems lists, video clips, redesign proposals or verbal briefings.

This description represents the ideal case. In reality, the interplay is more complicated. The design products may be unusable as a basis for evaluation and they are available too late in the development process. The evaluation process often takes too long, and the results seem to have a very limited effect on the design process.

The literature on HCI does not provide help on this problem. The HCI field includes a rich variety of techniques for either usability

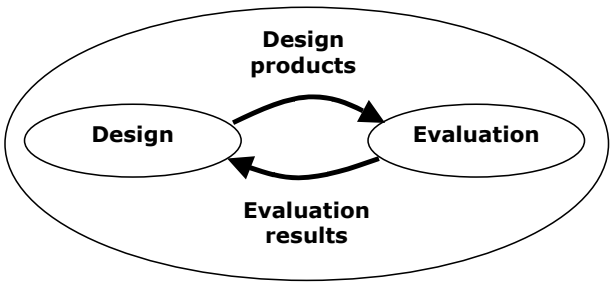

Figure 1. The interplay between user interface design and usability evaluation as key activities in software development

evaluation or user interface design. But there are very few methodological guidelines for the interplay between these key activities. In addition, there are no systematic surveys of research in this area.

3. Goal of the Workshop

The goal of the workshop was to determine state-of-the-art in the interplay between usability evaluation and user interface design and to generate ideas for new and improved relations between these activities. The aim was to base the determination of the current state on empirical studies. Thus authors were asked to employ empirical studies as a basis for presentations of new ideas on how to improve the interplay. Within this focus, the following topics of discussion were suggested:

- Which products of user interface design are useful as the basis for usability evaluations?
- How do the specific products from user interface design influence the techniques that are relevant for the usability evaluation?
- In which forms are the results of usability evaluations supplied back into user interface design?
- What are the characteristics of usability evaluation results that are needed in user interface design?
- Do existing evaluation methods deliver the results that are needed in user interface design?
- How can usability evaluation be integrated more directly in user interface design?
- How can usability evaluation methods be applied in emerging techniques for user interface design?

4. Overview of Papers

Ten papers were accepted for the workshop. They are divided into the following four groups:

A. Case studies of design and evaluation

B. User centered design (UCD)

C. Impact on Software Development

D. Reframing the Problem

Group A includes three papers that present results from empirical studies of the interplay between user interface design and usability evaluation. Kadytė and Tétard describes how usability evaluation was conducted and which usability testing techniques that were employed in the development of a mobile system. The usability evaluation helped the designers get a better understanding of the consequences of choosing different design options.

Murphy et al. reports from a project where usability evaluation was outsourced to an external evaluation organization. It is described how the evaluation process was structured, and the usefulness of different kinds of feedback from evaluation to design is discussed.

Paay and Kjeldskov presents the design of a prototype of an indexical context-aware mobile system. It is described how an understanding of the context of use is useful for planning usability evaluations.

Group B includes two papers that inquire into the extent to which the user centered design approach provides a way of handling the interplay between user interface design and usability evaluation. Venturi focuses on the extent to which user centered design techniques are used in the software industry, particularly in combination with the Rational Unified Process (RUP). Two patterns of integration are described and the challenges of integration are discussed.

Lárusdóttir presents a research plan for comparing the waterfall model with a user centred design approach. The research is based on student projects, and guidelines for these are also discussed.

Group C includes three papers that focus on key aspects of usability evaluation. Frøkjær and Hornbæk deals with feedback from evaluation to design. They have conducted interviews with designers in order to determine elements of feedback that are particularly valuable. They conclude that redesign proposals as opposed to mere problem lists are very valuable for software designers.

Skov and Stage inquire into the challenges of integrating usability evaluation into the design process by having designers conduct usability evaluations. A simple introduction to usability engineering is outlined and the results from teaching this to novice evaluators are presented. It is concluded that the novices became capable in some areas of usability engineering, but in others they still lacked competence.

Law deals with effectiveness of usability evaluation methods. Based on data from usability evaluations, it is discussed to what extent the problems identified induce fixing. It is also discussed more generally what effectiveness of a usability evaluation method is.

Group D includes two papers that provide a reframing of the topic. Hvannberg focuses on the relation between elicitation and design and between design and evaluation. The discussion is based on two case studies. It is suggested that design and evaluation are run concurrently in the development process with two related models as repositories.

Cockton argues that user interface design and usability evaluation both have to be placed within a value-centred framework. Usability evaluation deals with interaction, not designs. A value-centred approach is motivated and outlined; with that approach most of the questions raised in the call for workshop papers are reframed or rejected.

5. Challenges discussed at the workshop

To us, five challenges discussed at the workshop reading the interplay between evaluation and design stand out. They concern (1) the form and content of feedback from usability evaluation to user interface design; (2) achieving an early interplay between evaluation and design; (3) improving commitment towards and understanding of HCI and usability evaluation; (4) methodological problems in the research on usability evaluation and user interface design; and (5) challenges imposed by changing contexts of software development.

First, an important challenge concerns the *form of feedback* given from evaluation to design. Typically, user interface designers receive as feedback a report, listing usability problems with their design. However, several participants at the workshop argued that this form of output is problematic because the problems in the

report are often very short, too numerous, detached from the context in which they arose, and hard to understand. In addition, it is doubtful whether listing of problems are a key concern in actual software development. Previous research also suggests that not all problems raised in such reports are equally important; some problems may lead designers to waste time, should they try to correct them. Yet, research examining alternative forms of output from usability evaluation is rare.

Second, achieving *early interplay* between evaluation and design was identified as a key challenge. In particular participants agree that rescue HCI, that is late and cosmetic impact of evaluation on design, was unsatisfactory. Rescue HCI, however, seems to be happening a lot in software development. While this role of HCI in design to some extent may be the fault of HCI professionals themselves, the challenge to have early and value adding influence on the design of products nevertheless remains. The key here is to make usability evaluation be a part in shaping what gets designed.

Third, getting an *understanding* for how HCI may contribute to the software development have proven to be challenging; getting *commitment* to early and continuous focus on usability evaluation is even harder. These challenges include managing expectations of software designers, and being clear about what (and what not) HCI can do. Improving the relation between management and HCI professionals in particular, seems important: reward structures and top-level support on HCI are rarely in place. Several participants argued that too often management or designers hold unrealistic expectations, causing a sure-loss situation for usability evaluation and its interplay with software design.

Fourth, a number of *methodological* challenges were discussed, including the core issue of how to assess the ability of usability evaluation methods to impact user interface design. In particular, several participants questioned the reliance upon think aloud testing as a gold standard against which to assess alternative usability evaluation techniques. Another issue concerned how to ensure the validity of the usability issues identified with a product – while much research has produced usability evaluation techniques that can find many usability problems, little research have documented that those problems are real, let alone have useful impact on user interface design. Finally, many techniques and measures of HCI emphasize task-related performance measures, for example task completion times or accuracy. As products and services that we want to evaluate are increasingly dealing with experiences, games, and long-term interaction, we need to find better measures of subjective experience in order to, for example, make these criteria of iteration. However, especially linking those measures to design proposals seems hard.

Fifth, recent changes in *software development contexts* were discussed – for example diminishing time to market, faster development cycles and new devices. These challenges appear in practice to impose many constraints on the interplay between usability evaluation and user interface design. For example, the faster development cycles mean that less time is available for the actual evaluation, quicker analysis is needed, and more clear-cut advice is needed. Usability evaluation techniques and tools for these contexts are lacking.

6. Solutions Discussed at the Workshop

While challenges were numerous and easily describable, solutions were sketchier. Below we describe some of them.

One recurring suggestion was for more *empirical studies* of industrial scale design projects, thereby raising our understanding of the interplay between design and evaluation as it unfolds in practical projects. The focus of such studies could include how developers assess and chose to correct usability problems, the impact of various form of problem descriptions, and the evaluation of different representations of design, say use cases compared to paper prototypes. Such studies could also serve as exemplar case studies to be used in establishing realistic expectations of how usability evaluation and HCI could impact design. Initial explorations in this direction were presented by Lárusdóttir, Frøkjær and Hornbæk, and Law.

Another key idea was to strengthen the *coupling of evaluation and goals/values of the design*. All too often, evaluation is done with a too shallow understanding of the goals and values to be embodied by the design; evaluation also is done too late to matter. Several position papers presented ideas on how to feed information from design activities into the evaluation activities, for example through value statements and testable design rationales.

As evident from the section on challenges above, much more research is needed on the *various form of feedback* in which the results of usability evaluation is presented to developers. Such forms include redesign proposals, video highlights, and workshops. All of these have been at least initially explored with interesting results; however, studies examining the impact and persuasiveness of various forms of feedback are needed. In particular, the needs and wants of stakeholders in the design process should be carefully considered in relation to finding suitable and persuasive forms of feedback.

A supplement to the above ideas is to improve *evaluators' skills*. Little research has aimed at improving in concert the finding, analysis, filtering, and reporting of problems. The basic idea presented by Skov and Stage was to circumvent the gap between evaluators and developers by teaching basic evaluation skills to developers.

While the focus of the workshop was on empirical studies, the position papers made it plain that further work is needed before any clear solutions to improving the interplay between evaluation and design are reached.

References

Attewell, P. (1994), Information technology and the productivity paradox. In D.H. Harris (eds), "Organizational Linkages: Understanding the Productivity Paradox". Washington, DC: National Academy Press.

Beyer, H. & Holtzblatt, K. (1998), Contextual design, Morgan Kaufman Publishers

Brooks, Jr., F. P. 1987. No Silver Bullet: Essence and Accidents of Software Engineering, IEEE Computer, 20, 10-19.

Bødker, K., F. Kensing, and J. Simonsen (2004): Participatory IT Design. Designing for Business and Workplace Realities.

Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000), "Measuring usability: are effectiveness, efficiency, and satisfaction really

correlated?", Proceedings of CHI 2000, 345-352 ,The Hague Netherlands:ACM Press.

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

Greenbaum, J. & Kyng, M. (eds.) (1991), Design at work: cooperative design of computer systems, Lawrence Erlbaum Ass.

ISO (1998), "Ergonomic requirements for office work with visual display terminals (VDTs)-Part 11: Guidance on Usability".

Juristo, N., Windl, H., & Constantine, L. (2001), "Introducing usability", IEEE Software, 20-21.

Landauer, T. K. (1995), The trouble with computers: usefulness, usability, and productivity, MIT Press.

Lewis, C. (1982), "Using the "thinking-aloud" method in cognitive interface design", Research Report RC9265.

Mathiassen, L. and Stage, J. (1992) The Principle of Limited Reduction in Software Design. Information Technology & People, 6(2-3):171-185.

Nielsen, J. & Mack, R. L. (1994), Usability Inspection Methods, Wiley and Sons Inc.

Strassman, P. A. (1985), Information Payoff: The Transformation of Work in the Electronic Age, New Canaan, CT: The Information Economic Press.

The role of usability evaluation and usability testing techniques in the development of a mobile system

Vaida Kadytė
Åbo Akademi University
Turku Centre for Computer Science
Lemminkäisenkatu 14 B, 20520 Turku, Finland
Tel: +358-2-2153335
vkadyte@abo.fi

Franck Tétard
Åbo Akademi University
Institute for Advanced Management Systems Research
Lemminkäisenkatu 14 B, 20520 Turku, Finland
Tel: +358-2-2153350
ftetard@abo.fi

ABSTRACT
One characteristic of mobile application development projects is short time-to-market. Short time-to-market implies that very little time is available to application developers between the conception phase of an application and its actual implementation and launching. In the meanwhile, many activities should be conducted, including user requirements elicitation and analysis, application design, testing and evaluation. Along with these activities, a number of decisions will be made, which will influence the design of the user interface. In this paper, we focus on the use of usability testing techniques, and how these influence the design of the user interface in a mobile application development project. We make an account of a usability test, the techniques used, and the results obtained. The paper elaborates on these results with a discussion on how the use of usability testing techniques has influenced the project further on; this discussion is supported by an interview and comments gathered from a technical leader of the development team.

General Terms
Design, Human Factors.

Keywords
Usability testing methods, user interface design, mobile systems

1. INTRODUCTION
Application development of mobile systems is a growing industry. Application development of mobile systems has its own specificities, in the sense that features such as personalization, localization need to be implemented. This means that customer needs need to be understood, and therefore that a user-centred approach should be adopted in order to successfully fulfil user expectations. Also, one specificity of mobile application development is that there are many types of devices: mobile handsets are continuously coming with new additional features to entice users to upgrade, on the other hand, high-speed data capabilities through next-generation cellular networks (2.5G and 3G) trigger the demand creation of more sophisticated mobile phones. This variety of devices means that it is increasingly difficult to develop applications, which will work optimally on all devices. Moreover, mobile application development projects are characterized by short time-to-market, which means that very little time is available to application designers and developers, as well as project management between the conception phase of an application, its actual implementation and launching. In the meanwhile, the application development process should include activities such as user requirements elicitation and analysis, application design, testing and evaluation, to name a few. Along with these activities, a number of decisions will be made, which will influence the design of the user interface. Such decisions encompass e.g. choice of the optimal device for the application, the type of interaction style, or the type of interface. In this paper, we focus on the use of usability testing techniques, and how these influence the design of the user interface in a mobile application development project that was conducted together with a corporate client. We make an account of a usability test conducted during the project, the techniques used, and the results obtained. The paper elaborates on these results with a discussion on how the use of usability testing techniques has influenced the project further on; this discussion is supported by an interview and comments gathered from the technical leader of the development team of the project under scrutiny.

2. PROJECT DESCRIPTION
This paper is based on a research project which aimed at developing mobile business applications for a fine paper value chain. The fine paper production industry is a very mature one, where a long-term relationship with business customers is of particular importance. Recently it has been forced to focus on high quality and innovative products and ability to provide customer care with the help of new ICT capabilities. The project organization involved the third largest fine paper producing company in Europe and one of its key customers - the largest printer in Finland. The initial project goal was to design a mobile system that would provide access to the information services via mobile devices at the point of need, and consequentially would

also benefit B2B relationships in the fine paper value chain. The feasibility study including customer requirement elicitation was conducted in the form of action research, when a team of four people at a research organization participated in the actions of the target organization as consultant body. From the beginning of the application project in March 2003 until the final product was delivered in June 2004, the potential users from both companies were constantly involved in the process of interface design and had influenced its major features during monthly project meetings and workshops.

Even though both companies were very modern in terms of technology and investments in research and development activities, the application development project was a subject to various resource constrains. First of all, we had to decide on a mobile interface design for corporate users - senior and middle level managers - who were both unfamiliar with the mobile business applications and could not afford to spend much of their time on training and evaluation. What was really clear was that they urgently needed to upgrade their existing mobile phones to be able to do more than manage personal contacts and calendars. The business users were on demand to have wireless access to corporate e-mails and databases as well as the capability of running third party applets and services. Smart phones is a new class of handhelds that combine a mobile phone, MP3 players, camera and colour screens with integrated PDA functionalities (calendar, address book, to-do lists), and even the capabilities of running custom applets and accessing corporate databases and currently appears to be the hottest segment of the handheld market. Today these devices not only meet current business needs, but also provide a wide selection base in terms of design, sophistication of features and price, which also makes the usage and usability of mobile applications more complex by different users. In our research project we had a quite challenging goal - to develop a mobile product navigator for novice business users and decide on what kind of smart phone the system would run. Furthermore, the general condition was, that a new corporate standard - the selected smart phone and mobile application - would be easy to learn and usable. Further in this paper, we focus on the use of usability testing techniques, and elaborate on how these influenced the design of the user interface in a mobile application development project.

3. USABILITY TEST DESCRIPTION

3.1 Test objectives

Our first test-goal was to measure the impact of the two different designs on the prospective user (see Figure 1). In particular, we wanted to measure:

- how easy it is for a novice user to learn to use the system on a mobile device
- how easy and efficient it is to operate
- end-user attitudes towards the system

The second test goal was to identify specific problems that the user encountered with the design proposals and with the two devices. We wanted to measure functionality of the systems and users performances within the two system designs on two devices and we wanted to find specific problems that were associated with the usability of the mobile system. Our testing of the interface designs was not concerned with the separate components of the system but concerned more the combination of the components so that we could evaluate how "user-friendly" or good for the purposes the chosen design and device was. We expected that the results of the usability test would help the design team in making decisions regarding further design of the system, and help us giving recommendations whereas which device and which type of interface should be favoured when the system will be taken into use.

Figure 1. Graphical Screen Interface (Grid view) and Hierarchical Screen Interface (List view)

The following general guidelines were set up for the usability test:

1. General questions:
   - On what design is it easier for the user to find the right information?
   - Which design is more intuitive to use?
   - Which design is easier to learn?
2. General goals:
   - A system that is easy to learn and use
3. Quantitative goals
   - Find best performance on device and design, the fastest alternative/user
   - The user should make no errors
4. General concerns
   - Is the system pleasant and easy to use for the purposes it is intended
   - Is the system logical to the actual end-users (e.g. is the menu structure in accordance with their understanding of the product groups)?

The guidelines mentioned above were operationalised as a number of performance measures and subjective measures. Performance measures were directly linked to the quantitative usability goals as well as to the general concerns that were driving the usability test (Dumas & Redish, page 189). Subjective measures, such as opinions, perceptions and judgments of end-users, were linked to the general concerns of the test; these measures were partly operationalised as scores using the PANAS scale (see 3.2), and as qualitative data collected from post-test interviews and by asking users to think aloud during the test(Dumas & Redish, page 187).

The following performance measures were used:

- Time needed to complete a task.
- Number of errors per task.
- Ease of learning: time difference in task completion between two same tasks.

The following subjective measures were used:

- Ease of learning the system.
- Ease of using the products.
- Attitudes towards the system.

3.2 Usability testing methods

We used several data collection methods during the test. Having several data collection methods would ensure that we get evaluation insights of different kinds, which we would feed further into the design process.

User tests: User tests can be conducted in several ways: in the user's natural environment (e.g. on-site testing) or in a controlled environment (e.g. laboratory testing). Both approaches have advantages and disadvantages: choosing the proper environment is very often a matter of trade-off which must be assessed in respect with the objectives of the intended test. In our case, we chose to conduct a laboratory test for two main reasons: (i) we had easy access to laboratory facilities, and (ii) the intended test users, i.e. managers, were willing to join a test in a laboratory. The advantages of running a test in a laboratory were that we could easily control, record, and measure the interaction of the users with the system. Laboratory test planning and data analysis are time-consuming activities: this can be considered to be the main disadvantage when organizing such a test.

User comments: User comments were collected in two different manners. First, users were urged to think aloud when using the system: the think-aloud method is commonly used, although it can be argued that (i) it may distract the users for the task at hand and that (ii) not all users are eager or able to express their thoughts during the course of a test. Second, in-depth interviews were conducted after the test; the purpose of these interviews was to collect feedback mainly about ease of learning the system, and attitudes towards the system.

PANAS scales: PANAS (Positive and Negative Affect Schedule) mood scales have been developed to measure positive and negative affects of individuals. Positive affect (PA) reflects the positive feeling, the extent to which a person feels alert, enthusiastic and active. A high PA means the person is in the state of high energy, full concentration and pleasurable engagement (Watson *et al.*, 1988). A low PA stands for e.g. lethargy and sadness. The Negative Affect (NA) stands for unpleasurable mood states like anger, disgust, guilt, fear and nervousness. A low NA means the person is in a state of calmness and serenity (Watson *et al.*, 1988). The PANAS scales developed by Watson *et al.* enable us to measure PA and NA as two distinct uncorrelated dimensions of affective structure. The PANAS scale is generally known for its stability and it is a relatively easy and trouble free method, short and quick to administer. Some positive and negative affects have been found to be related e.g. to satisfaction and social activity, self-reported stress and poor coping (Watson *et al.*, 1988). We used the PANAS mood scale in order to determine if one of the designs was likely to cause more positive and/or negative feelings than the others.

3.3 Test user selection

The basis for selecting users was that their profile would be as close as possible to the intended user population. For this reason, we aimed at testing managers of the project company and one of their clients. These managers would be somehow literate in computer use, and be familiar with a basic business phone (however, not necessarily with the most advanced business phones available at the time when the test took place). An important selection factor was that they needed to be familiar with the business and the jargon used in the business under scrutiny; this was important as it would ensure that users do not spend time questioning, for example, the meaning of different product groups during the test.

The actual test users of the mobile system are middle level managers. Three of them work at paper producing company (with activities such us product marketing, sales and customer service) and four of them - at a printing company (business customer of the paper company, whose employees are involved in purchasing the paper, warehousing, production planning and control). What unifies the test users is that they all work within the same value chain of fine paper products; they are aware of the complexity of those products and know each other organizational processes very well. On the other hand, the test users have different job focus and responsibilities: four of them have a more business oriented job role (users 1,3,5,7) and three of them are more characterized as technical people (users 2,4,6).

Their average work experience is quite extensive (over 10 years) and indicates that test users are very familiar with the content available for purchasing and processes associated with it. They can be called experts of the content, which is to be navigated via mobile device. Pre-test questionnaires revealed interesting results about their knowledge: since most of the test users know the product information by heart, they have almost never used the product navigator in an electronic format even though it has been available on the web for the last few years. This sort of design of usability test works quite well for the selected fragment of the value chain, because the paper company and printing house operate in a local Finnish market, which is simple and pure and business between them is based on trust and long term relationship.

4. EMPIRICAL RESULTS

The usability test conducted during the project enabled us to gather a lot of data about the users' interaction with the system. We used Noldus usability testing software for recording and analysing observations. As it is not the scope of this paper to make a detailed account of the usability test results, but rather to reflect on the usefulness of usability testing methods and results for interface design, we will present only general statistics and the main insights of the test.

4.1 Data from Observations

Table 1. Observational Data Based on Quantitative Measures

Interface\Device Combinations & Tasks		Task Duration (seconds)		Number of Errors	
		M	SD	M	SD
Joystick List View (JL)	1	122.66	70.90	3.29	3.99
	2	58.10	31.43	0.29	0.49
Joystick Grid View (JG)	1	109.81	98.08	1.14	2.27
	2	84.04	47.73	0.86	1.57
Stencil List View (SL)	1	74.17	25.99	3.00	2.77
	2	38.63	19.03	1.43	2.57
Stencil Grid View (SG)	1	102.73	55.02	1.57	1.27
	2	53.40	24.30	1.29	1.38

Note. SD = Standard deviation, M = Arithmetic mean (n = 7)

Three main insights can be derived from the data presented in Table 1:

- Learning of the user: There is a learning effect taking place. Users perform better (both in terms of task duration and number of errors) task 2 than task 1, and this independently of the device and type of interface.
- Different performance depending on the device: Users perform better when using a device using a stencil rather than a joystick as interaction mode.
- Different performance based on the type of interface: Users perform better using a list-based interface rather than a grid-based interface.

4.2 Mood States with PANAS Moment Instructions

In general PANAS can reflect a general mood state which is fairly similar among the test participants (see Table 2).

Table 2. Positive and negative mood states based on PANAS

	U1	U2	U3	U4	U5	U6	U7	M
Positive Affect (PA) mood states								
JL	2.5	3.6	2.7	2.5	3.4	2.6	2.9	2.89
JG	2.5	3.5	2.5	2.3	3	2.4	2.9	2.73
SL	2.3	3.5	2.6	2.5	3.2	3	3.1	2.89
SG	2.3	3.6	2.4	2.6	3.3	2.5	3.1	2.83
Negative Affect (NA) mood states								
JL	1.3	1	1	1	1	1.2	1	1.03
JG	1	1	1	1	1.3	1.2	1	1.11
SL	1	1.1	1	1	1	1.2	1	1.04
SG	1	1.2	1	1.1	1	1.1	1	1.06

4.3 In-depth Users' Interviews

In-depth interviews were conducted with each user after the test. The main findings of the post task interviews are summarised in Table 3.

To us observers it first seemed that user 1 had more difficulties with using the joystick phone, but the interview and the analysis show that he found this phone easier to use. He found the icons to be moderately descriptive and was of the opinion that for him the picture memory works, you get used to the pictures easily and then you remember them.

User 2 reported few problems with the touch screen phone concerning sensitivity of the stencil, but it was better than the joystick. The results agree with his comments, since he used less time by using the stencil phone. He said that he did not really think about if the icons were descriptive enough but if the system will work in the same way as a normal system on your PC you can organize your desktop so that you have the information you want under your own icons, which is useful. The results show that this user was faster by using the grid layout.

User 3 made a point about the different appearance of the two phones at a meeting with customers. He said that the user would look more professional with the stencil phone at a meeting, because it would look like he would sit and write something, which looks more professional than scrolling on a phone with a joystick looking like you are playing with your phone. The test user preferred icons because he believed icons will be "the thing of the future".

Table 3. Summary of Post-tasks Interviews

	JL	JG	SL	SG
U1	Difficult to get information overview.	⊕	The stencil phone left him feeling unsure, if it will respond to a pen touch or not.	
U2	It is slower than the stencil phone, the screen is smaller		'Too much of scrolling down to find the information' - he was about to give up.	⊕
U3	'It is slow, the screen size smaller and does not look for professional use'.		List view would be far too long – their product list is growing steadily.	⊕
U4	±	Not "his thing", too small, "unresponsive" and "difficult"	±	Would be worried of losing a pen. Can't navigate with one hand.
U5	±	±	±	±
U6	'Information column gets so long that you cannot really use it on the phone'.	⊕	'A pen was a bit misleading: how hard or soft it needs to be pushed; for a correct vertical scrolling – weather to push or to drag a scroll bar; too tiny and can be lost'. Can't navigate with one hand.	
U7	Phone is reacting slower, looks bulky; information bar is not descriptive enough.		List does not look nice	⊕

Note. ⊕ = User expressed clearly his/her preferred combination of interface & device; ± = User doesn't have a clear preference and doubts for corresponding combinations.

User 4 could not reflect his preference for any phone. He stated in the interview that the joystick was not "his thing", he felt it was too small, "unresponsive" and "difficult", even though he did not have any apparent trouble using it. The touch pen appeared to be quite natural for him to use, but at times he got stuck in a situation where he pressed too softly, then too strongly, and became frustrated and clicked a few more times to no avail. In the interview he stated that he felt also the touch pen to be a bit difficult. He also said he would be worried of losing it. He felt that the joystick phone was slower than the stencil, but still he performed his fastest task with the former. User 5 was concerned with the logic arrangement of information in the system and the icons meant "nothing" to him.

Navigation on the phones was somewhat difficult and not intuitive for user 5. Especially the joystick created difficulties for him as he did not seem to find a comfortable way of operating it and slipped quite many times. He also experienced errors when using the touch pen; especially he would hold the pen for too long

on the surface instead of quickly tapping. This made a new menu pop up, which he managed to handle quite independently though. When using the joystick, the screen backlight turned off several times while he was thinking of his next view, which meant that he had to move the joystick in some direction to put the lights back on. This he felt to be a major irritation and something he would really be very annoyed with in the long run. He did not like the icons very much and felt that pictures represented nothing meaningful for him. Otherwise he said, he would like icons, since he is familiar with that metaphor from the computer world.

According to user 6, the joystick phone feels better in the hand than the touch screen alternative. The former is designed so that it should be he held in ones palm. And in deed user 6 behaved a bit more bravely with a joystick phone and immediately took it into his palm.

User 7 liked the touch screen phone at the first glance and didn't change her mind after the test. It was a larger screen size and elegant navigation with a stencil, which gave such a positive impression on her. She also mentioned that a joystick phone didn't look different from other phones she saw, and the joystick itself reminded her a bit of computer games what she never plays.

5. DISCUSSION AND CONCLUSION

The usability test conducted during the project helped the project team to gather a lot of information about how various design options would influence the performance of intended users of the system. It also helped the project team to collect user comments which could be used for further improvements of the system. In the following, we will aim at summarizing the results obtained, the experience gained from using several usability testing methods, and how the results actually influenced the remaining of the project. The discussion presented here is supported by comments of the technical leader of the project; these comments were collected during an informal interview conducted after the project was completed.

Statistical data collected from the observations show clearly that (i) there is a learning effect taking place (comparisons of task 1 vs. task 2), (ii) a list view is more effective than grid view, and (iii) most of the users perform faster with the stencil based navigation than with joystick based navigation. Concerning the benefits of the usability test, we could say that, although we gathered significant and tangible results, the organization of such a test is time and resource-consuming and similar results can probably be obtained with more informal test methods. The main benefit of a usability test remains in the opportunity it gives to the usability experts, development team including technical leader, and project manager all together to monitor many test factors and review these later after the test is conducted.

PANAS was generally better for monitoring the overall mood states of the users during the test: the test results did not reveal any significant individual differences over the positive or negative affects that the different solutions had on the users. As a tool, PANAS with moment instructions was not very useful, since it didn't bring any concrete and reliable insights. Only in three out of seven user cases, PANAS mood states correspond with the users' statements from the post-task interviews. Without in-depth user interviews, interpretation of PANAS results would be difficult and even dangerous.

Post-task interviews with the test participant resulted in rich and subjective information regarding different type of interfaces and input mechanisms. It revealed that three out of seven users had more positive preferences towards using a stencil phone, due to its relatively larger touch screen, faster reaction and elegant look. Two users expressed their clear preference for the phone based on joystick navigation as this kind of method it seemed to be more reliable and more familiar to them. However, the other two users, after listing all the drawbacks for each phone, could not come to the conclusion which one they would prefer. In terms of navigation designs of a product catalogue on a smart-phone, the majority of test users preferred a grid view. We considered unsatisfactory comments from user 4 and 5 concerning a bad organization of information content for the frequent users and suggested an alternative for them – an advanced search was built into the navigator.

User comments collected during the test did not seem to be very helpful in terms of suggestions for design improvements, as they were mostly related with general questions and concerns regarding the phones' features. Use of the think-aloud method proved to be unsuccessful for two main reasons: (i) subjects were not trained to think aloud, (ii) as experts of the system content, users were somehow familiar with the task at hand, and thus they could not easily verbalize their actions, (iii) all of the test participants were Finnish and by nature inherited many characteristics of a high-context culture.

The results of the test are interesting in the sense that user interviews do not necessarily validate the statistical results obtained: it is clear that users perform better using a given device and a given interface, but, surprisingly, these results are not obvious to the users, as they were unable to agree on the best device and the best interface (most users preferred the interface with which they performed worse). The technical leader confirmed that he was surprised that users preferred the grid view to a list view, which proved to be more effective during the test (Technical leader's comments: *"Yes. I did not expect most of the users to prefer the grid view with colourful icons. Another thing is that even if most of the users were performing better and less confused with the list view, they still liked the icon (grid) view more"*). These results did not influence the actual interface design during the project, but the technical leader reflected on how he would overcome this design problem in future designs (technical leader's comments: *"I still believe that you should try to avoid placing pictures or symbols. I believe it is hard for people to associate a certain symbol with particular thing: they have to learn it first, and with time it may be useful to navigate information in such a way. Personally, I do not like to remember symbols, numbers or pictures. I remember the combination and certain placement of information in a certain way. Therefore next time I would develop a product navigator with a grid view, but instead of picture icons, I would have single colour icons with a text below. I also realized that the user interface with icons is very handy with a stencil - you have a larger area and in most cases you will succeed to touch in the right place"*). The technical leader of the project commented that the test results were somehow expected, but did not think that these would influence further design of the system (technical leader's comments: *"Not really (influence). (I) expected most of the responses and anticipated it during the design phase. Different people like different mobile devices as well as webpage layouts. … I also was*

not surprised that people like the stencil phone more since it is faster to navigate: you see and you touch, no need to go up, down or to the left. But of course in certain situations a joystick phone is very handy – you can navigate the phone only with your thumb").

Our technical leader expressed the need for further testing of the system (technical leader's comments: *"It would be interesting to test the ordering system, which is now integrated within the product navigator. The more complex system you have, the more interesting it is to test. Our product navigator has very little functionalities"*).

To summarize, the test proved to be useful to validate several design choices made by the technical leader of the project. From a pure design perspective, the system under scrutiny proved to enhance user performance, although users' opinions and comments seemed to suggest that a less effective design would be preferred. Also, results collected with PANAS moment instruction unfortunately were not confirmed with the interviews; therefore we did not take into account the PANAS scales in our data analysis and decision making.

6. REFERENCES

[1] Dumas J., and Redish J. *A Practical Guide to Usability Testing*. Intellect Ltd. 1999, p. 189.

[2] Watson D., Clark L., and Tellegen A. Development and Validation of Brief Measures of Positive and Negative Affect: the PANAS Scales. *Journal of Personality and Social Psychology*. 1988, Vol. 54, No. 6, pp. 1063-1070.

Location, location, location: Challenges of Outsourced Usability Evaluation

John Murphy[1], Steve Howard[2], Jesper Kjeldskov[3] and Steve Goschnick[2]

[1] Design4Use
7 Abbotsford Street
Melbourne, Victoria 3067
Australia
john@design4use.com.au

[2] Department of Information Systems
The University of Melbourne
Parkville, Victoria 3010
Australia
{showard, stevenbg}@unimelb.edu.au

[3] Department of Computer Science
Aalborg University
DK-9220 Aalborg East
Denmark
jesper@cs.auc.dk

ABSTRACT

This position paper presents some of the challenges experienced in relation to an outsourced usability evaluation of commercial collaboration product, which we would like to raise in the Improving the Interplay between Usability Evaluation and User Interface Design workshop. The paper describes the context of the outsourced evaluation, three challenges of location, how the evaluation was carried out and reported. Finally, we outline some of the lessons learned.

INTRODUCTION

A commercial company is developing a new product, which is intended to support collaborative work amongst non-technical commercial workers. For this product to succeed, non-technical users must be able to use the tool easily. A significant component of the ease of use of the product is the users' ability to create a clear and coherent mental model of the system. In order to evaluate the current design of the product, it was decided to conduct a usability evaluation of the current design. The overall objective for the usability test was to determine whether the product supports a coherent and consistent mental model for a user collaboratively sharing files with others to achieve a goal. The secondary goal of the evaluation was to determine whether the interface screen design and flow supports the individual tasks of creating and sharing through the product with another person and accepting an invitation to share.

The evaluation allowed us to study some of the challenges of outsourcing usability in a large industrial software development project. In the following sections, we first briefly introduce the context of the product usability evaluation. Secondly, we outline some of the challenges encountered in planning and conducting the evaluation, which we would like to address in the workshop. Hereafter, we briefly describe how the evaluation was carried out and the mechanisms employed for reporting them. Finally, we outline some of the lessons learned.

BACKGROUND

The product is being developed within a multi-national software product company based in the United States. Typical of this type of company, the product company has a multitude of existing and new products under development in various programs under aggressive time and resource constraints. The company has a strong commitment to being focused on the needs of customers in relation to their products and services. As such, the company has strong human computer interaction (HCI) skills supporting the development of user interfaces that are easy to use. However, the number of these resources is limited in relation to the number of projects and amount of HCI work required. As with many companies throughout the world, this product company is investigating an outsourcing model to support HCI requirements and in particular usability evaluation.

The company has offices in Australia that, aside from day-to-day business are involved in HCI based research in collaboration with the Universities of Melbourne and Aalborg. This program has been running for over four years encompassing collaboration on developing research techniques, industry projects, teaching and sponsorship of a state of the art usability laboratory in The University of Melbourne, Department of Information Systems. The university has strong skills and resources in usability evaluation and is very active in research and teaching of evaluation techniques.

Given this collaborative relationship, the company decided to pilot a set of usability tests on one of their developing products using the Company – Melbourne – Aalborg relationship. Following discussions with senior company product managers, the product was selected as a suitable candidate based on it being at an appropriate state of development, requiring HCI support and being an open source development which circumvented non-disclosure requirements.

From the company perspective, the objective of the testing was to determine whether cost effective useful findings

could be established through timely testing (as discussed in e.g. Kjeldskov et al. 2004). This required timely setting up of the software, designing the test, recruiting participants, running the sessions and analyzing and reporting of results. The design of the testing had to be determined appropriately with the knowledge that development was continuing throughout the testing period and the testing should be budgeted to be cost effective relative to running the testing in the United States.

CHALLENGES TO THE EVALUATION OF THE PRODUCT

Usability testing and evaluation faces challenges: some generic and some features of the particularities of the evaluation under question; some interesting and others mundane. In this section we focus on three challenges that we found particularly problematic: location, location and location.

Location – Geography

Conducting a remote usability evaluation places a particular burden on communication and the maintenance of situation awareness (Murphy 2001, Hartson et al. 1996). Multiplexed time zones can aid in rapid turn around of results but only if synchronous interaction is not required at times of unavailability, or indeed uncivilised hours, and only if the disparate teams are 'talking the same language'.

Prior to commencing the evaluation, and drawing on a mix of local knowledge, documentation, email and teleconferencing skills, we harvested as much understanding of the remote situation as we were able. In the workshop we will discuss the influence that the following had over the project:

- Expectations on rapid turn around time and streamlined reporting requirements
- Preferences for and bias toward different data collection methods and data types
- Concern that usability evaluation produce more than merely a list of problems (i.e. the results should be translated into design change suggestions)
- Interest in the process (how the evaluation was conducted) as opposed to merely the product and the findings from the evaluation.

Location - Sector

Combining multiple sectors (in this case industry practitioners and university researchers and research students) is a real strength of our approach. The established and ongoing relationship between the company and the Universities of Melbourne and Aalborg allows us to respond rapidly to emerging opportunities under the rubric of a tested agreement. However, as a cross sectoral collaboration it is not without its frustrations (but see Lambert, 2003 for some solutions). Some of the issues we will raise in the workshop include:

- The need for industry partners to be able to guarantee short cycle delivery times whilst recognising the imperative that university researchers' engage is risk oriented longer-term discovery.
- The need for university based researches to balance consulting and applied research with more basic enquiry.
- The importance of exposing PhD students to 'real world' projects whilst at the same time limiting unnecessary distractions to their ongoing thesis work.
- The management and protection of intellectual property; both background and created intellectual property of the researchers, the students and the industry partner.
- Gauging the benefits that flow from any collaboration, be they immediate and tangible or more speculative.

Location – Development phase

Usability evaluators, be they located in industry or universities, are unfortunately rather experienced at being introduced too late into the lifecycle to have a major impact on the product. It was therefore rewarding to be invited to comment at a relatively early stage in a product's development (see Rubin, 1994 for a discussion of the importance of life cycle positioning). However, an opportunity to comment early should not be confused with an occasion for unbridled creativity! Some of the issues we should like to raise in the workshop include:

- Gauging the degrees of freedom available in responding to the identified usability flaws.
- The critical importance of the representational form of any feedback to the design team.
- Balancing a critical perspective on the present design with a constructive account of the next.

Faced with these challenges of outsourced usability evaluation, we designed and conducted an evaluation of the product in collaboration between the company and The University of Melbourne and reported the results back to the development team in the United States. The design of the usability evaluation and the way we reporting back the results are described below.

EVALUATION DESCRIPTION

The product usability evaluation was conducted over two days at a state-of-the-art usability laboratory at The University of Melbourne, Australia. The evaluation was done in a collaborative working environment with real life scenarios and tasks requiring the use of other software such as e-mail client and folder and file manipulation tools. Two independent usability evaluations were conducted; a user-based evaluation and a heuristic walkthrough. These are described in detail below.

User- Based Evaluation

The user-based evaluation was based on the think-aloud protocol, involving three triads of test subject working collaboratively through the product. The test subjects were physically separated from each other and could only collaborate using the product and e-mail.

Each of the three evaluation sessions took approximately one hour and consisted of a collaborative task requiring the three users to share information by creating, sharing and using the product. During the evaluation, the subjects were presented with a scenario and tasks to complete.

The scenario was based on the common financial task of sharing and updating work plans within a finance group. This scenario was selected as common across many companies and performed by staff requiring no particular technical knowledge. The profile of the test subjects were non-technical knowledge workers who, ideally, could be part of a team who are used to working together. The subjects were not employed by the company and did not have any special knowledge of the company software.

The user-based evaluation sessions were recorded on digital video capturing all overviews of all three test subjects and their respective computer monitors.

Heuristic Walkthrough

Secondly, three Doctoral students specializing in Human-Computer Interaction conducted a Heuristic Walkthrough of the product software using the scenarios described above.

The Heuristic Walkthrough session lasted approximately ninety minutes and was facilitated by the first author who recorded usability problems by the expert reviewers for later analysis and comparison with the user based data.

REPORTING THE RESULTS

The evaluation had several audiences - project stakeholders in the form of product managers and senior product development staff, company HCI professionals based in the United States and most importantly, product engineers actually working on the product. Each of the different audiences required different information; the project stakeholders were most concerned with the feasibility of outsourced usability evaluation in terms of costs, resources and overall effectiveness; the HCI professionals were concerned to validate the evaluation process and results to both ensure the quality of the results for the product work, but more importantly to investigate how and whether this process and resource might be able to support on-going company HCI work; and the product engineers wanted "design ready" results. From a product engineering perspective, it was understood that the reporting of problems would not be useful without some accompanying proposal of a solution, particularly in the case of significant or complex problems.

Given these different audiences and reporting requirements, a number of different reporting mechanisms were employed. A telephone conference was used to report high level findings, costings and an overall project feasibility to stakeholders and HCI staff. A short highlights video of the usability laboratory, equipment and 'snippets' of the actual evaluation was prepared to present the evaluation process to the company HCI staff and stakeholders. A written evaluation report was prepared explaining the results in detail for product engineers and company HCI staff. It was structured with a usability problem summary table, a discussion of each of the usability issues, user interface design solution ideas and a description of the test.

The evaluation results were well received by the company in the United States. The cost of running the evaluation was within budget and is believed to be a cost effective opportunity for the company. Further investigating into the outsourcing model is currently in progress.

LESSONS LEARNED

The product software is still under development and prone to errors. These factors led to a significant increase in the standard level of support and intervention required for usability testing. For instance, participants required support where the ability of a user was significantly different to the other team members and needed to maintain timely collaboration with colleagues. In cases where participants acting as team leaders sharing files and becoming entangled in Microsoft file-sharing were assisted back to the product environment to maintain the flow of the task. Also, it was important that users were not distracted and did not spend significant cognitive effort on things such as learning an unfamiliar e-mail client or manipulating folders.

In relation to the process of evaluating the product, a strong background contextual knowledge is essential to ensure the testing is effective. Budgets, timelines for product development intended audience are all used to support the design of the evaluation. Other deeper and more subtle knowledge such as market share for this product, future plans to integrate with other products, main competing products and number and skill of engineers available to work on the product are just a sample of the broader knowledge that is useful in supporting the design of the testing.

The physical setup of hardware and software environment and skilled technical support for a product in development is also a challenge. For example, one of the product requirements was a static IP address which was not able to be obtained in the University environment. The company engineers in Australia spend one full man day and University technical staff spent almost half a day setting up the environment and software. This challenge may also be viewed as an advantage in the enforcement of independence through at all levels based on the remoteness of the testing. Not only are the evaluators and evaluation staff independent, but also the entire technical setup is required

to be independent which may in itself reveal technical system 'bugs'.

The video highlights were found to be extremely valuable as a fast effective mechanism of providing a significant amount of information to the project stakeholders and company HCI staff. The video highlights viewed in conjunction with the teleconference meant that the presentation and ensuing discussion quickly became informed and focused.

The value of this type of work to the researchers is in exposure to real world systems, provision of actual data for their research and provision of money to support their facilities. Real world systems expose researchers to actual problems and systems serving to ground their thinking and research ideas in reality. This also applies to usability laboratory staff who broaden their experience in setting up and running real industry evaluations. Providing commercial confidentiality can be preserved and data can be made suitably anonymous this work can be a source of actual data for research projects. This can be implemented through establishing a suitable commercial reviewing process. Finally and significantly, researchers benefit from the money that is directed towards maintaining the quality of the University evaluation facilities and staff.

ACKNOWLEDGEMENTS

We thank the test subjects who participated in the the product evaluation and the Doctoral students who conducted the heuristic walkthrough; Jeni Paay, Sonja Pedell and Jan Skjetne.

REFERENCES

Hartson, H., Castillo, J., Kelso, J., Kamler, J., & Neale, W. (1996) Remote Evaluation: The Network as an Extension of the Usability Laboratory. Proceedings of CHI'96. http://www.acm.org/sigchi/chi96/proceedings/papers/Hartson/hrh_txt.htm (accessed 8/26 2004)

Kjeldskov, J., Skov, M. B. and Stage, J. (2004) Instant Data Analysis: Evaluating Usability in a Day. Proceedings of NordiCHI 2004, Tampere, Finland, ACM, pp. 233-240

Lambert (2003) Lambert Review of Business-University Collaboration http://www.hm-treasury.gov.uk/media/EA556/lambert_review_final_450.pdf (accessed 8/26 2004)

Murphy, J. (2001) Modelling 'Designer – Tester – Subject' Relationships in International Usability Testing. IWIPS 2001 Designing for Global Markets 3 July 12-14 Milton Keynes, United Kingdom Editors, Donald L. Day and Lynne Dunckley

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

Evaluating Indexicality: The Importance of Understanding Place

Jeni Paay[1] and Jesper Kjeldskov[2]

[1] Department of Information Systems and
Faculty of Architecture, Building and Planning
The University of Melbourne
Parkville, Victoria 3010
Australia
jpaay@unimelb.edu.au

[2] Human-Computer Interaction Group
Department of Computer Science
Aalborg University
DK-9220 Aalborg East
Denmark
jesper@cs.auc.dk

ABSTRACT

The research presented in this workshop paper proposes the importance of understanding users' perception of a "place" when designing and evaluating the usability of mobile indexical information systems. It is our contention that by acquiring a detailed understanding of social and physical aspects of a built environment through empirical field studies, we can explore people's ability to make sense of their surroundings in the design of interfaces for context-aware mobile information systems. In addition, we argue that the understanding of a place developed through such field studies can also play an important role in informing the planning and conducting of subsequent evaluations. Supporting this, we present a field study conducted at Federation Square in Melbourne, Australia, and the design of an indexical context-aware mobile prototype. We then discuss some challenges and benefits associated with using the experience from conducting the field study to inform not only the design of the prototype but also the planning of our forthcoming usability evaluation.

1. INTRODUCTION

Many mobile information systems involve the user being situated in the context of public built environments. Yet only a few studies have investigated the challenges imposed and opportunities offered by adapting these mobile information systems to the context of buildings and other architectural structures in urban spaces. In order to exploit the user's ability to make sense of architectural features in the physical surroundings in interaction design for context-aware mobile information systems, we need to achieve a better understanding of the role of the user's physical environment in defining their context and the contribution of existing information embedded in that environment to people's experience of it and to their situated interactions (Agre 2001, Bradley and Dunlop 2002, McCullough 2001). Also, we need to learn how to make a clear connection between the user's physical surroundings and the information presented on their mobile devices (Dix et al. 2000, Persson et al. 2002).

1.1. Indexical Interfaces for Mobile Devices

An interesting approach to making a clearer relation between mobile device interfaces and the user's context is to apply the idea of *indexicality*. Indexicality is a concept drawn from semiotics, which is currently being applied to the design of mobile device interfaces to streamline the information and functionality delivered to the user (Kjeldskov 2002, Paay and Kjeldskov 2004). In relation to interface design, indexicality is defined as a property of a representation that has a context-specific meaning and thus only makes sense in a particular context. The idea of applying indexicality to mobile human-computer interaction is that if information in the interface can be *indexed* to the user's situation, then information already provided by the context becomes implicit and does not need to be displayed. Hence, the user's environment becomes part of the interface. On the basis of this, the limited screen real estate of mobile devices can be optimized to contain only the most vital content.

In order to include meaningful and useful indexes to the built environment in context-aware mobile devices, the key properties of the target built environment needs to be understood and modeled. Subsequently, examples of indexical interface design need to be carefully evaluated. So far, our research has resulted in the development of systematic methods for (1) gathering, analyzing and understanding the properties of built environments that provide insight into the user's physical and social contexts (Paay 2003) and (2) creating analytical abstractions of this data, in the form of descriptive frameworks, which can be used for informing mobile device interaction design. On the basis of this, a prototype design has been developed and is currently being implemented (Paay and Kjeldskov 2004). We are now faced with the challenge of how to evaluate context-aware indexical mobile device interfaces in order to provide designers with appropriate feedback on the validity of their understanding of the built environment being designed for and the usability of their specific interface design. This is the topic we would like to discuss at the workshop.

In the following sections, we briefly describe the field study conducted and outline some of the outcomes including the design of our prototype system. After this, we reflect on the lessons learned from our initial field studies and discuss how these can inform the planning and conducting of our forthcoming usability evaluation.

2. FIELD STUDY: EXPLORING PHYSICAL CONTEXT

In order to investigate the role of the user's physical environment and the contribution of existing information embedded in that environment to people's experience of it, we conducted a field study of the recent architecturally designed Federation Square, Melbourne, Australia. Federation Square is a multi-modal public space with a mixture of distinct architectural features and embedded digital elements that provides a variety of activities to visitors. The aim of the field study was to identify important properties of the built environment as an inhabited public space and to create an analytical abstraction, which could inform the design of a mobile information system supporting visitors to this place.

The field of architectural design has a history of incorporating user needs into design methods for the built environment. Urban Planner, Kevin Lynch (1960) and Architect, Christopher Alexander (Alexander et al. 1977) have both modelled built environments, specifically cities, with regard to the people that inhabit those places, hence implicitly including the users in their analysis of physical space. Their methods have not only proven their value within architecture, but have also been applied previously to human-computer interaction design (see e.g. Dieberger and Frank, 1998, Borchers 2001).

Combined with qualitative research methods, the work of Lynch (1960) and Alexander et al. (1977) inspired the development of a method for analyzing and modelling the architectural and social elements of a physical place for the purpose of informing human-computer interaction design for mobile or pervasive information systems. In the method devised, observational expert audits were made of Federation Square, recording through photographs and field notes the elements of the physical environment for concept formation and open coding analysis. Encoding schemas based on classifications from both Lynch (1960) and Alexander et al. (1977) were used to combine elements of the images with observational field notes and classify them. Open coding was then used to identify critical terms, key events and themes and to derive categories that synthesized the outcomes of this empirical study.

2.1. Outcomes from the Field Study

The key outcomes from the analysis phase were (1) a map identifying four key districts and four key landmarks, and (2) an abstracted visualisation of an emergent "vocabulary" of the human experience of the informational and architectural properties of the space called MIRANDA (Multilayer Information Related to Architecture aNalysis Data Abstraction). This is illustrated in figure 1.

MIRANDA clarified and identified the essence of the characteristics of the space, providing a representation of human experience of that space. Surveying the resulting diagrams (as illustrated in figure 1), it is possible to draw summary conclusions about the space, which would not be evident from viewing the original data, or from merely visiting the space. For example, at a glance it is possible to ascertain that Federation Square has a dominant characteristic of "Activity around the Edges" indicated by the dominant line linking the two words.

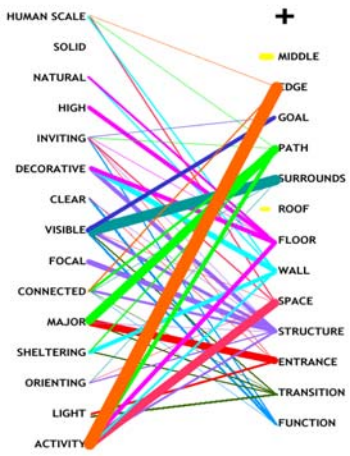

Figure 1. Abstraction of the human experience of Federation Square based on MIRANDA.

3. AN INDEXICAL MOBILE PROTOYPE SYSTEM

Exploring the use of MIRANDA in the design of physically indexed interfaces for mobile devices and making way for an empirical user-based evaluation of indexicality as a design concept for context-aware mobile information systems, we have designed a mobile information system for use at Federation Square. The design is currently being implemented as a functional prototype using Bluetooth for positioning and GPRS for wireless access to the Internet. The prototype system incorporates three overall design ideas exploiting unique characteristics of the physical space analyzed and indexing to some of the identified features of the built environment:

- The mobile guide responds to the user's location in terms of one of the defined districts rather than Cartesian coordinates;
- Each district is represented in the mobile guide by an interactive photorealistic depiction of the physical surroundings augmented with textual or symbolic information needed to better understand the place;
- Locations and instructions for navigation are expressed through rich descriptions derived from the distinctive characteristics of the place rather than through Euclidian coordinates.

Four districts define the user's location. This acknowledges people's ability to make sense of the physical environment in which they are situated, and that location is not defined by coordinates but by the human experience of the physical layout of the space. The location districts and the corresponding screens are illustrated in figure 2.

The information pushed to the mobile device is tailored to information needs within a specific district. When a user moves into a new district, their context changes, and so does the information that appears on the screen of their mobile device. As the user enters a district, an interactive photorealistic depiction augmented with textual and symbolic information pertinent to that district is pushed to their device. To allow the user to align the information on their display to their physical surroundings, the initial screen displays the corresponding landmark for that district. From this starting point, the user can alter the current perspective and select linked information.

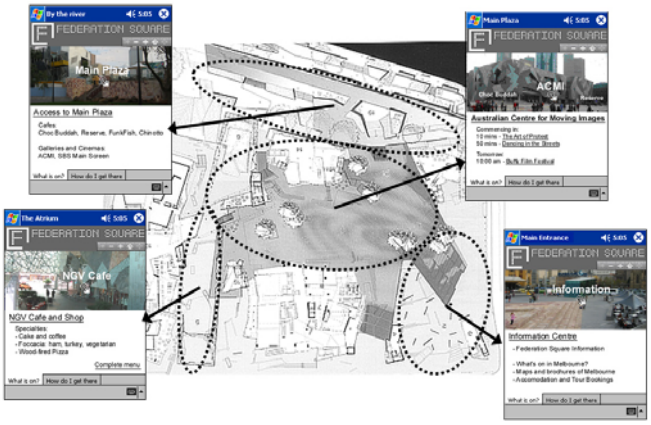

Figure 2. Four location districts and corresponding screens.

Based on the knowledge from MIRANDA we are able to use terms in the rich descriptions that relate to human experience of the space. This is an alternative to absolute location descriptions typically used in mobile guides, and holds more meaning to the users of the system, because it makes use of their understanding of the built environment in which they are situated, thus indexing the information in the interface to the user's physical environment.

4. EVALUATING INDEXICALITY

The discussion of whether to conduct usability evaluations of mobile devices in laboratory or field settings is ongoing (see e.g. Kjeldskov and Stage 2003). For the prototype system described above, however, we believe that a field evaluation at Federation Square will be needed to investigate the usability of aspects of the interface design that rely fundamentally on indexing to the user's built environment. Even though several studies have documented the possibility and value of simulating use contexts in laboratory settings, we believe that this will not be possible to do satisfactorily with the built environment of Federation Square.

Conducting usability evaluations of mobile devices in the field is, however, not trivial. Mobile use contexts are often highly dynamic and involve several physically distributed actors. Also, field evaluations complicate data collection and limit means of control. In relation to the evaluation of indexical interfaces for context-aware mobile systems targeted at public spaces, these challenges are taken to an extreme. Public spaces are typically very crowded and lively and the subsequent analysis requires that the collected data provides views of (at least) 1) the interface display and user's interaction with it, 2) the user's perception of his physical settings, and 3) how the user is situated in, and interacts with, objects and people in the physical space surrounding that person.

Planning and conducting an evaluation that meets these challenges can be very complicated, time consuming and difficult to get right. However, having conducted considerable field studies in the use domain prior to the evaluations has provided us with valuable insights.

4.1. Lessons Learned From the Field Study

Apart from informing the design of our prototype, the time spent in the field during the collection of empirical data for the MIRANDA framework also resulted in significant experience with conducting field studies generally and at Federation Square specifically. It has also provided us with detailed knowledge of the elements of the built environment itself and with anecdotal evidence of people's interactions in it. In turn, this has provided us with important input for planning and conducting the forthcoming usability evaluation of our prototype system.

One of the key lessons learned from the empirical field study leading to the prototype development was the need for flexibility and adaptability in relation to original study plans. Even the best-laid plans can go awry and be difficult to execute when encountering unexpected conditions, such as bad weather, huge crowds attending special events, or large temporary structures. Since participants are not easy to reschedule, the field study typically has to go ahead despite changed conditions, often requiring the investigators to improvise and make impromptu decisions.

A second lesson was to fully understand the limitations of our data recording equipment in different conditions, and to have strategies ready for collecting the best possible data under difficult conditions. During the analysis phase we found that high quality sound was vital but that wind, traffic and crowds all interfere significantly with this. To get a good sound recording, the camera needs to be directly in front of the speaker, which is difficult when participants are on the move and when you don't want to lead them in a specific direction. Also, people tend to talk less in crowded spaces, and thinking aloud in public spaces seems to make participants feel socially uncomfortable. In our present study, we also found it impossible for the interviewer to take field notes on paper while on the move.

4.2. Some Implications for Usability Evaluation

To date there has been no empirical evaluations of indexical mobile information systems. Our prototype therefore provides a unique opportunity to test the applicability of this concept in interface design. Unlike many other mobile information systems, the proposed design explicitly uses insight into user perceptions of the built environment to tailor the information presented on the screen to the users physical context. In our evaluation we thus have to investigate (1) the validity of the underlying analysis of the outcome from the field study, (2) the success of transferring knowledge about the place encapsulated in MIRANDA into a system design, and (3) if the indexes between the interface and the user's surroundings are accurate, meaningful and effective.

The following guidelines for our evaluation have emerged from our understanding of the place investigated and the challenges experienced when conducting fieldwork there.

Planning and flexibility. Careful planning is needed to find ways to efficiently use the time spent in the field to collect relevant data. The time available for a single visit is determined by tape time, battery time, and human enthusiasm (which observably waned after 2 hours in the field). In data collection this was best achieved by using any stationary time for reflective contextual interviews. An overall plan, or checklist, that covers all aspects that need to be tested during the visit allows participants to use their individual paths through the space and ways of doing tasks, but the evaluator can adeptly guide them to complete all tasks within the planned time limit, and thus minimise activities that do not contribute to the data.

Participant briefing. A familiarity session with participants before going into the field is needed so that they fully understand their part in the evaluation. More importantly, they need to practice "think aloud" protocol while being video taped with others watching them, (which is not natural to most people), *before* they go into the field. Although Federation Square is a place where tourists are often videoing and photographing each other, participants still found it difficult to think aloud.

Recording equipment. Movement around the square is useful because it tends to trigger conversations, but recording speech while on the move is difficult. Multiple camera angles are needed including a camera directly in front of the participants to facilitate lip reading during transcription. An improved technique for recording conversations, such as radio microphones should be used, and there should be periods of time where the participants are drawn to a quiet location and asked to reflect on what happened in the previous evaluation task. Rehearsal of the use of all the simultaneous recording equipment under difficult circumstances (such as e.g. while holding an umbrella) would also be a useful exercise.

We believe that a situated familiarity with a specific context, gained through fieldwork during analysis and design phases of mobile system development, better supports the evaluation of indexicality in interface design for mobile systems.

ACKNOWLEDGEMENTS

Thanks to Steve Howard and Bharat Dave for supervision. The field study was designed and conducted by the first author. The prototype was designed in collaboration with the second author supported by the Danish Technical Research Council (ref. 26-03-0341 and 26-04-006).

REFERENCES

Agre, P., 2001, Changing Places: Contexts of Awareness in Computing. *Human-Computer Interaction*, **16,** 177-192.

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-king, I., and Angel, S., 1977, *A Pattern Language: Towns, Buildings, Construction* (New York: Oxford University Press).

Borchers, J., 2001, A Pattern Approach to Interaction Design. (Chichester, England: John Wiley & Sons, Ltd).

Bradley, N., and Dunlop, M., 2002, Understanding Contextual Interactions. Proceedings of Mobile HCI 2002 (Pisa, Italy: LNCS, Springer-Verlag), pp. 349-353.

Dieberger, A., and Frank, A., 1998, A city metaphor to support navigation in complex information spaces. Journal of Visual Languages and Computing, 9, 597 – 622.

Dix, A., Rodden, T., Davisen., Trevor, J., Friday, A., and Palfreyman, K., 2000, Exploiting Space and Location as a Design Framework for Interactive Mobile Systems. *ACM Transactions on Computer-Human Interaction*, **7(3)**, 285-321.

Kjeldskov, J., and Stage, J., 2003, New Techniques for Usability Evaluation of Mobile Systems. International Journal of Human-Computer Studies (IJHCS) Elsevier, 60(2003):599-620.

Kjeldskov, J., 2002, Just-in-Place: Information for Mobile Device Interfaces. Proceedings of Mobile HCI 2002 (Pisa, Italy: LNCS, Springer-Verlag), pp. 271-275.

Lynch, K., 1960, *The Image of the City*. (Cambridge, Mass.: MIT Press).

McCullough, M., 2001, On Typologies of Situated Interaction. *Human-Computer Interaction*, **16**, 337-349.

Paay J., and Kjeldskov J., 2004, Understanding and Modelling the Built Environment for Mobile Guide Interface Design. To appear in Journal of Behaviour and Information Technology

Paay, J., 2003, Understanding and Modeling Physical Environments for Mobile Location Aware Information Services. Proceedings of Mobile HCI 2003 (Udine, Italy: LNCS, Springer-Verlag), pp. 405-410.

Persson, P., Espinoza, F., Sandin, A., and Coster, R., 2002, GeoNotes: A Location-based Information System for Public Spaces. Proceedings of Mobile HCI 2002 (Pisa, Italy: LNCS, Springer-Verlag), pp. 151-173.

Integrating the User Centered approach in the design of Command and Control systems

Giorgio Venturi
Thales Group – Netherlands
Haaksbergerstraat 49 – 7554 PA – Hengelo, The Netherlands
giorgio.venturi@nl.thalesgroup.com

ABSTRACT
We conducted an enquiry on the usability practice of different industries in order to discover the most powerful strategies in implementing the User Centered Design (UCD) process. Most important factors are sharing the usability goals with the customer, considering UCD as a business strategy, using UCD in competitive analysis and communicating UCD values outside of the company. Analysing our situation we have started building up a baseline of usability requirements, specific to our task domain, which can improve the negotiation between the customer and the supplier of the systems and consequently lead to a better integration of UCD within the company.

Author Keywords
Computer Human Interaction, User Centered Design, UCD Integration, Command and Control, UCD Survey, Usability Requirements.

INTRODUCTION

Within Thales Naval Nederland (TNNL) UCD has been applied for 4 years as an iterative, model-based process in the design of the man-machine interface of command and control systems. The process employed is a tailored version of Usage Centered Design [1], particularly focused on the modeling of the tasks and of the interaction. The former system has been redesigned through this process and now provides a better support to the work of the operators.

Anyway, we are still not satisfied with the current UCD implementation into the company. We considered first to define and assess our process through a capability maturity model.

Capability maturity models (CMM) have been employed for more than a decade to assess the maturity of the software/system engineering process; a number of CMM have been proposed specifically for the HCD/UCD processes [2,3] as well. While these reference models are valuable for process assessment and process definition, we wanted to understand how to evolve our position within the company. What are the strategies that other UCD practitioners put into practice? What are the obstacles that they must face? We decided to design a web-based survey [4] in order to discover which are the most common obstacles and strategies in implementing the UCD approach[1].

SURVEYING THE UCD PRACTICE

Definition of the Sample

Research sample includes UCD practitioners in the industry, spanning from large companies and corporations (Computer, Financial, Telecommunications, etc.) to small, specialized consultancies. We gave the communication of the web survey via e-mail, to the major newsgroups and forums related to usability and UCD (ACM-SIGCHI, IDX, UK-usability, BCS-HCI among the others). 83 practitioners successfully completed the web survey in a time frame of 40 days. Most of them are human factor specialist (34%) or user interface designer (33%) and have between 5 and 13 years of UCD experience, with a minimum of 1 and a maximum of 45 years. They come from different business sectors, with most of the companies following two patterns:

1. Big companies with more than 1000 employees;
2. Small sized (<50), independent usability consultancies.

The first pattern

According to the first UCD integration pattern, the concentration of UCD practitioners in a company is comparable to a drop into the ocean: on the average 2-3 practitioners over 1000, less than 1% of the total number of employees of the company. Moreover, UCD activities are still mainly funded through the R&D budget (48%), much more than bill-back by projects (36%) and annual budget (31%): this means that UCD is still seen as research, not incorporated into the mainstream processes.

How many years ago was UCD first applied? Most of the companies have started applying it since a not so short time, between 2 and 6 years ago (Figure 1): in the same timeframe in which the RUP got a grip in the software industry, UCD has barely put down its roots on it.

[1] The questionnaire and the raw data are not attached due to space limitations, but they are available on request from the authors.

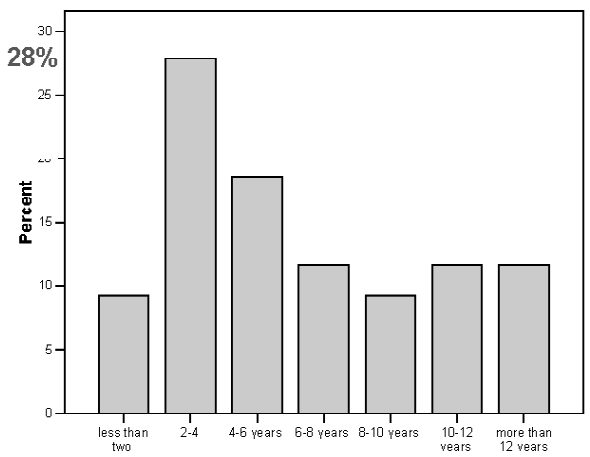

Figure 1. How many years ago was UCD first applied?

The second pattern

According to the second pattern, usability consultancies employ less than 50 people (100%) and have a high ratio of UCD employees instead. More than half of them are organized in teams and are funded at the project level.

The second pattern shows globally a very well integrated approach; this is not a surprise since UCD is their main business activity.

Manager commitment

What about the commitment of the managers? Here we got apparently contradicting figures: while 61 percent of them thinks UCD to be part of their business strategy, they usually do not set usability goals (only 25% do), nor do they usually compare the usability of their products to that of their competitors through competitive analysis (again, only about 40%).

It seems that, when applied, UCD is mostly considered as a selling proposition, without seriously incorporating it into the business of the company. As a result, when we face an economy downturn, usability funding is cut, as if it was "unnecessary luxury".

Most used methods

In the survey we asked also what kind of methods and techniques have been employed in a chosen, representative project.

Prototyping is obviously the most used approach during the design phase (Figure 2), in its low-fidelity and high-fidelity variants. An interesting trend is the substantial similarity between the two figures of the low-fidelity and the high-fidelity approach: the low-fi prototyping is more used in the analysis phase, while the high-fi in the design phase. Some years ago the low-fi variant scored much higher [4], which is due probably to the improvement and/or the release of new prototyping tools. Prototyping is quite often coupled with formative, qualitative usability testing (about 60%).

In the evaluation phase (Figure 3), observation and formative usability evaluation still score quite high, while summative, quantitative usability evaluation scores only 27%.

Expert and heuristic evaluations are much less used today (38% during design and 33% during test) than some years ago [6], where they were used by about 70% of the practitioners.

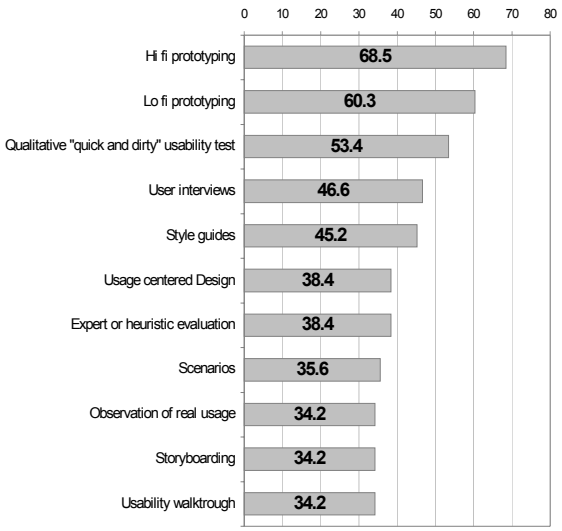

Figure 2. Methods most frequently used during the design phase

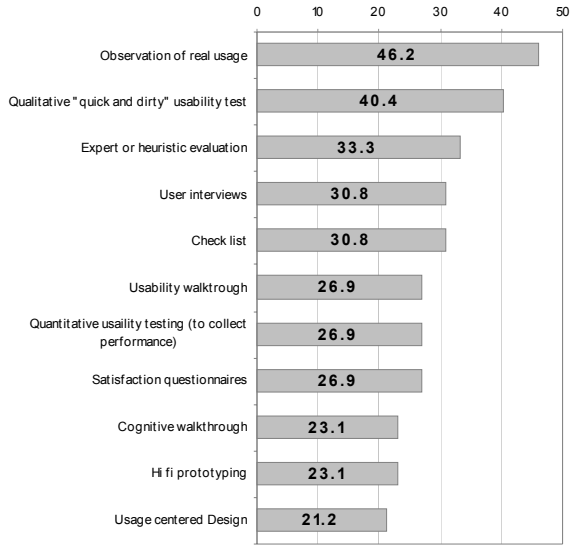

Figure 3. Methods most frequently used during the test phase

Overall, the "user interview" is the most frequently used method over the whole lifecycle, since about 80% of the surveyed UCD practitioners used it at least once.

The importance of UCD integration

In our study [4] the descriptive analysis shows a cluster of companies that are achieving success in the implementation of UCD. We applied therefore other types of analysis (ANOVA, factorial) in order to select the most relevant factors. The most relevant set is made by "sharing the usability goals with the customer", "UCD as business strategy", "UCD in competitive analysis" and "Outbound communication". All of those factors are significantly related to the number of practitioners in the company and the budget spent in UCD activities.

The importance of integration is therefore very high in the achievement of UCD benefits. While the process model and the UCD skills and knowledge are often available, like in our case, the factors related to the management, the infrastructure and the communication of usability are otherwise underestimated.

DISCUSSION

The findings of this study cast under a different light our current implementation of UCD. While we employ experienced professionals and an up-to-date process models, our approach is still lacking from the point of view of integration: the number of UCD practitioners is low, usability requirements, if defined at all, are a source of conflict with the customer, UCD is not part of the business strategy, UCD is not used in competitive analysis and outbound communication is barely carried out. Our integration level is therefore low; we can predict that on a Usability CMM assessment we would score for most of the practices at the first or second maturity level ("initial" or "managed").

Some of those problems, like lack of competitive analysis cannot be solved in our domain, because it is difficult to compare our command and control systems with those of different companies, while the communication can be easily addressed putting more resources into it. Improving the sharing of the usability goals with the customer, instead, requires more effort to be solved.

The open issues

Started in the innovation department, the UCD approach got progressively positive feedbacks from the programs management and at the moment it is more funded by programs than by R&D budget.

Anyway, there are three open issues to be solved yet:

P1. In projects there is often hardly any involvement of the customer in the domain modelling; the context of use is seldom used as guidance for the design and, as a consequence, it is impossible to define the usability requirements for the interface.

P2. It is very difficult or impossible to get feedback from the user after the product is deployed, unless the program clearly specifies it: usability tests are seldom employed in most of the military programs in Europe.

P3. The Usage-centered approach was accepted because it is fitting quite well in the whole Rational Unified Process and because it is founded on a structured analytical design process (Domain → Task → Interaction → Implementation) and therefore culturally close to the traditional engineering culture. Anyway, it does not really impact, as intended, the degree of user involvement in the design process.

Setting up a baseline for usability requirements

Through a number of internal interviews we found out that most of the suspicion towards usability is grounded in practical problems, common with other industries, expecially those that design and build safety/mission critical systems.

In our domain requirements are specified through a formal process, which involves the customer, the supplier, procurement agencies and research institutes. Specifying usability requirements can be tricky especially because requirements are later used in acceptance tests, and usability involves not only the capabilities of the system but also those of the team involved.

Requirements have a legal value and they specify the features of the system being delivered. But what if the system includes also the user? As suppliers, how can we avoid the risk of being rejected for the results of a usability test, which may go wrong because the team was not properly manned or trained?

Usability requirements bring different degrees of risk to the customer and to the supplier [7]: while performance measures ("*Expert user shall perform task Q and R in 5 minutes*") push the risk on the supplier, other requirements, at the design level ("*Systems shall use screen pictures in app xx, buttons work as app yy*"), as well as development

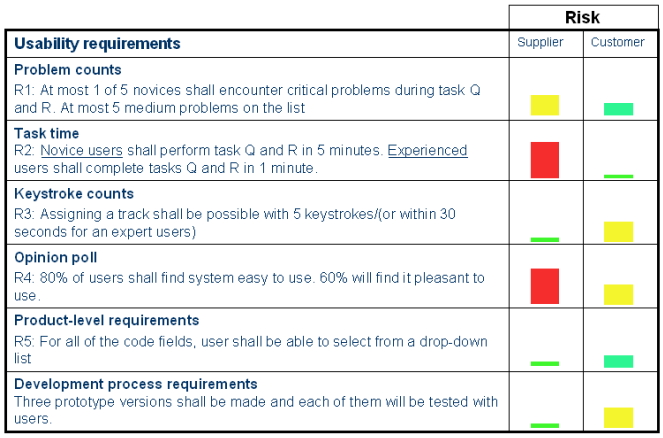

Figure 4. A baseline for usability requirements

process requirements ("*Three prototype versions shall be made and usability tested during design*") bring more risk to the customer, because they do not necessarily imply that a usable system is provided.

We started therefore to build up a solid baseline of usability requirements (examples in Figure 4) for our domain that can minimize the risk for the supplier and the customer and that can be used as a reference for future and existing programs.

CONCLUSIONS

We carried out a web enquiry in order to discover the obstacles and the most successful strategies in the implementation of the UCD approach. We found out that the integration factors are relevant in the achievement of a significant impact on the company business. In our company the integration of UCD is still modest: negotiation and definition of usability requirements was a critical point in the establishment of the UCD practice and it has been addressed first.

ACKNOWLEDGMENTS

This research has been supported by a Marie Curie fellowship of the European Community programme "Improving Human Research Potential and the Socio-economic Knowledge Base" under contract number HPMI-CT-2002-00221.

REFERENCES

1. Constantine, L. L. and Lockwood, L. A. D., Software for Use, Addison-Wesley, New York, 1999.

2. ISO/PAS 18152, Ergonomics of Human-system interaction – Specification for the process assessment, International Standards Organisation, Geneva, Switzerland, 2004.

3. Jokela, T., Assessment of User-Centred Design Processes As A Basis For Improvement Action. Doctoral thesis. Oulu University, Finland, 2001.

4. Venturi, G. and Troost, J., Survey on the UCD Integration in the Industry, NordiCHI04, Tampere, Finland, 2004.

5. Vredenburg, K., Mao, J.Y., et al., A Survey of User Centred Design in practice, CHI letters, Vol. 4, Iss. N. 1, 2002.

6. Rosenbaum, S., Rohn, J.A., Humburg, J.,. A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys, in CHI 2000 Conference Proceedings, 2000.

7. Lauesen, S., Software Requirements: Styles and Techniques, Addison-Wesley, 2002.

The Usability of a User Centered Design approach

Marta Kristín Lárusdóttir
School of Comp. Science, Reykjavik University
Ofanleiti 2, 109 Reykjavik Iceland
+354-599 6200
marta@ru.is

ABSTRACT
The usability of software will be extended, if developed by a User Centered Design approach. The drawbacks are not as obvious. This position paper describes a research plan for comparing the benefits and drawbacks of two software developing approaches, the traditional software development approach and a User Centered Design approach.

Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management – *cost estimation*, *life cycle*, *productivity*, *programming teams*, *software process models, time estimation*.

General Terms
Management, Measurement, Documentation, Performance.

Keywords
Software development approaches, feedback methods, user evaluation, document reviews, cost and benefit analysis.

1. INTRODUCTION
Decision makers in the industry ask: If I have $300.000 and I want to develop software for my company, how can you convince me, that your User Centered Design (UCD) approach gives me the most value for my money? How can I know that the UCD approach gives me a better product than the traditional one? These are very valuable questions and really hard to answer. A recent survey by Vredenburg et. al. shows that measurements of the effectiveness of the UCD approach are limited [4]. One of the conclusions in that survey is that there is a great need for common evaluation criterion for the UCD approach across industry.

So, what is a good criterion for measuring a software development approach? Are the criteria: a) the quality of the product developed; b) the experience when using the different approaches; c) the organizational benefits; d) the financial benefits e) or some other criterion? Could the ISO definition [2] of usability, function as quality criteria for measuring a software development approach, that is: Could the approach be measured according to the definition of usability by measuring the effectiveness, efficiency and satisfaction?

The UCD approach has been described in various details over the past decade or so, starting with Nielsen [6] to the resent ones, Mayhew [5], Preece et. al. [7] and Gulliksen and Göransson [1] to name a few. The ISO 13407 [3] gives a certain consensus for describing what the UCD approach is, but there the UCD approach is described from a higher level of abstraction than in most methodology books. Evaluation criteria for the UCD approach should fit the industry as well as the different methodological approaches.

This position paper describes a research plan for measuring the usability of two software development approaches, a UCD approach and a traditional software development approach. The research has been planned to start in January 2005 and has already been prepared.

2. THE RESEARCH PLAN
This section describes the goal of the research, the projects involved, the structure of it, the planned measurements and finally the methods used.

2.1 The goal
The goal of the research is to answer the question:

What are the costs and benefits of using a User Centered Design approach when developing software compared to the costs and benefits of using the traditional software development approach?

Measurements will be done on the effectiveness, efficiency and satisfaction for the two approaches.

The goal of the research is illustrated in figure 1.

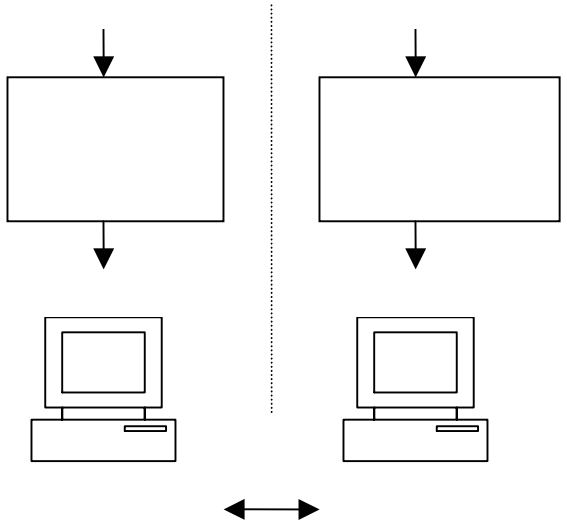

Figure 1: The goal of the research

During the same period of time, University students will develop software either according to a UCD approach or a traditional software development approach. In the UCD approach feedback on flaws in the analysis, design and programming is given to the students by concerning users, mainly through evaluating with users. In the traditional approach the students will get feedback on flaws from the customer or the mentor for the project through document reviews.

2.2 The software projects

Students in Computer Science do a complete software project as one of their final courses in their BS-degree studies. They usually work in a group of 3 people and get 12 ECTS points each for their work. Icelandic companies suggest the subject of the projects to the students and all the work is done at the company's site, where the students get all facilities and good connection to the customer and often the users, so these student projects are developed in somewhat real settings. In the following the students will be referred to as developers.

Usually these projects are 1.600 to 2.000 man hours running for five months with various subjects, one could be a plain CRUD (create, read, update, delete) project and another one could be more advanced, sort of a "proof of concept" project. No two projects have the same subject.

The data gathering in the research project will take two years and the estimated number of projects is 15 each year. The first year the developers will use a traditional software development approach but on the second year the developers will use an User Centered Design approach. Both approaches have the same milestones, delivering subprojects or documents with one months interval, see figure 2.

In the traditional software development approach the developers deliver requirements document, project plan and risk analysis during the first period of the project, design document during the second period and user and system manuals during the third period.

Finally the developers deliver the software developed and updates on all the documents on the delivery date.

All the documents need to be reviewed by the customer or the mentor for the project and a review summary will be made for each period of the project.

In the UCD approach the developers deliver the same documents during the first period of the project, but more focus will be on describing the users and their tasks than in the traditional approach. During the second and the third period the developers deliver prototypes that have been evaluated with users. For each period the developers deliver a summary of the user evaluations and comments.

The main difference of the two approaches is in the ways feedback is given to the developers, in the UCD approach users are contacted but in the traditional approach feedback is given to the developers through document reviews.

2.3 The structure of the research

As shown in figure 2, data will be gathered both during the process of developing the software and after the projects have been delivered. Five questionnaires will be used during the process, the first is mainly used to gather background information from the developers, the three iteration questionnaires will mainly be used to gather information on the methods used during that iteration and the developer's satisfaction. The final questionnaire will be used to gather information on the time used during the project and the developers overall satisfaction with the project and the applied software developing approach.

After the projects have been delivered, the quality of the outcome will be measured by user testing the projects with at least three users each. Furthermore the customer's satisfaction will be measured by using questionnaires and interviewing some of them.

The research will be running for three years, during the first two years the focus will be on data gathering, measuring the software development approaches during spring 2005 and spring 2006, but the last year will be concentrated on data analysis.

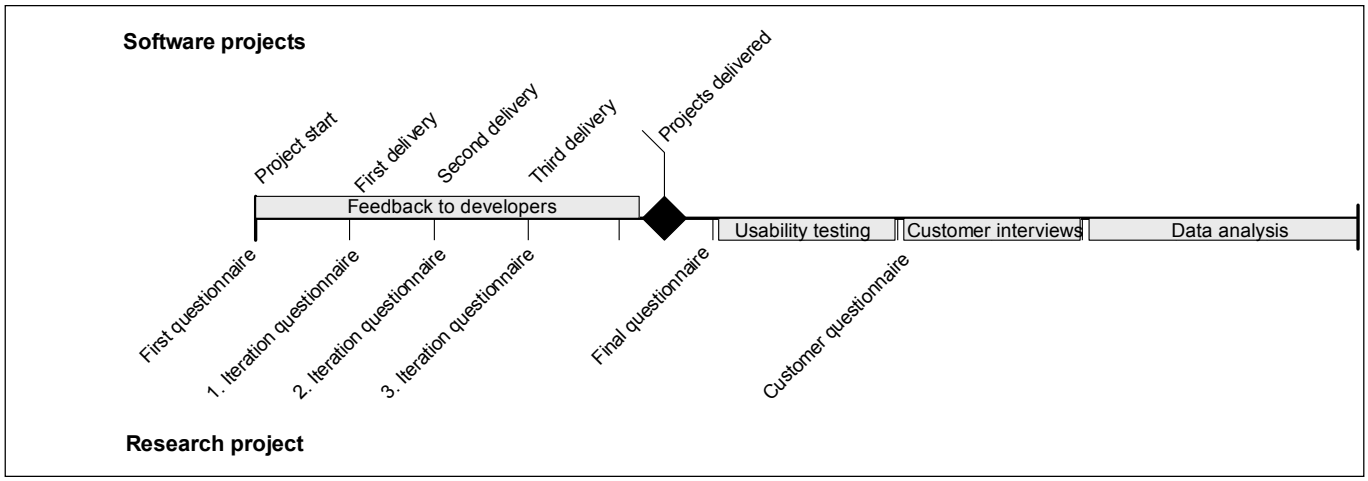

Figure 2: The proceeding of the student projects and the research project.

2.4 The measurements

The planned measurements are suited to gather information on the effectiveness, efficiency and satisfaction during and after using the software development approaches. In the following section, it is described what the planned measures are.

2.4.1 Measuring effectiveness

In the ISO definition of effectiveness [2] it is stated that: "Measures of effectiveness relate the goals or subgoals of the user to the accuracy and completeness with which these goals can be achieved". When measuring the effectiveness of getting feedback to the developers using a software development approach the collected data will be:

a) Was it manageable to get the feedback to the developers or not.

b) Number of problems found during the feedback gathering.

c) Quantitative measures on the quality of the feedback.

d) Quantitative measures on the quality of the product made.

2.4.2 Measuring the efficiency

Measures on efficiency are defined as [2]: "Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources". Expenditure of resources is measured by time used here, namely by:

a) The time used by the developers for getting the feedback.

b) The time used by the customer or users for getting the feedback.

2.4.3 Measuring the satisfaction

Finally, satisfaction is defined as [2]:"Satisfaction measures the extent to which users are free from discomfort, and their attitudes towards the use of the product." Here satisfaction will be measured by:

a) Quantitative measures on the satisfaction of the developers after using a particular method for feedback gathering.

b) Quantitative measures on the satisfaction of the developers after following the whole software development approach.

c) Quantitative measures on the satisfaction of the customer with the product developed.

2.4.4 Testing the planned measurements

All questionnaires for the research have already been made and tested during similar software projects during spring 2004. Many iterations were made on the questionnaires and interviews were made to gather information. At first the questionnaires were on paper, but the developers liked the electronic version better.

2.5 The methods

Three main data gathering methods will be used: questionnaires, interviews and acceptance testing. Additionally information on the feedback to the developers will be gathered. In figure 2 there is an overview of the schedule for the data gathering and in the following subsections the methods will be described briefly.

2.5.1 Questionnaires

The software projects are done in 4 iterations, each with one month duration. The questionnaires will be used to gather information on the developers and customer's satisfaction and collect descriptive data on what methods were used and how much time it took to used them.

2.5.2 Interviews

Some selected customers will be interviewed to get a closer look at their satisfaction. This will be semi-structured interviews.

2.5.3 Acceptance testing

The acceptance testing will be done by running user tests that the developers have prepared. All the tests will be run in the same location and by the same person to get as little bias as possible. Three real users of the systems will be asked to attend and a pilot test will be run. The results from the acceptance testing are very important to compare if the UCD approach results in extended usability of the software as stated before compared to the usability of the software developed by a traditional approach.

3. DISCUSSION

Being able to describe the costs and benefits of using User Centered Design approach with quantitative data and compare it to the costs and benefits of using a traditional software development approach will be a good tool in the fight usability people are having every day, when trying to convince customers and other software development people that keeping the focus on the users in the development of software is a fundamental thing for better quality of the software.

4. ACKNOWLEDGMENTS

I would like to thank all the participants in the workshop: *Improving the Interplay between Usability Evaluation and Interface Design* at NordiCHI 2004 for very valuable comments.

5. REFERENCES

[1] Gulliksen, J. and Göransson, B.: *Användarcentrerad design*, Studenterlitteratur, Lund, 2002.

[2] ISO/IEC. 9241-11 Ergonomic requirements for office work with visual display terminals (VDT)s – Part 11 Guidance on Usability, ISO/IEC 9241-11:1998 (E), 1998.

[3] ISO/IEC. 13407 Human-Centred Design Processes for interactive systems, ISO/IEC 13407: 1999 (E), 1999.

[4] Mao, J., Vredenburg, K., Smith, P. W., Carey, T.: User-Centered Design Methods in Practice: A Survey of the State of the Art*, Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research,* Toronto, Ontario, Canada.

[5] Mayhew, D. J. The *Usability Engineering Lifecycle*, Morgan Kaufman, San Francisco, 1999.

[6] Nielsen, J., *Usability Engineering*, Academic Press, Inc., San Diego, 1993.

[7] Preece, J. Rogers, Y., Sharp, H.: *Interaction Design,* John Wiley and Sons Inc., New York 2002.

Input from usability evaluation in the form of problems and redesigns: results from interviews with developers

Erik Frøkjær
Department of Computing
Universitetsparken 1
DK-2100 Copenhagen
+45 35321456
erikf@diku.dk

Kasper Hornbæk
Department of Computing
Universitetsparken 1
DK-2100 Copenhagen
+45 35321400
kash@diku.dk

ABSTRACT
Usability problems predicted by evaluation techniques are useful input to systems development; it is uncertain whether redesign proposals aimed at alleviating those problems are likewise useful. We compare problems and redesign proposals as input from usability evaluation into industrial software development, as discussed in the literature. We do so by presenting comments from interviews with system developers on what aspects of problems and redesigns they find to be of utility. Our study suggests that redesigns should be given more attention, both in comparisons of usability techniques and in practical usability evaluation.

Categories and Subject Descriptors
H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces—Evaluation/Methodology; D.2.2 [Software Engineering]: Design Tools and Techniques—User Interfaces.

General Terms
Measurement, Design, Experimentation, Human Factors.

Keywords
Usability evaluation, redesign, think aloud, metaphors of human thinking, empirical study, usability inspection.

1. INTRODUCTION
Most of the research on usability evaluation methods assumes that good usability evaluation techniques are those that best support an evaluator in generating problem descriptions while using the techniques; Hartson et al. [4], for example, suggests treating usability evaluation techniques as functions that produce problem lists, ignoring issues of how to treat problem descriptions and redesigns. This assumption has several limitations:

- Problem descriptions are sometimes very brief. The 46 usability problems described in [7, appendix 1], for example, is on the average about 28 words long. Therefore, problem descriptions may appear unclear or incomprehensible to readers other than the evaluator.
- When analyzing the effectiveness of usability evaluation techniques, problems are often compared in order to match similar problems. This matching process, however, turns out to be difficult and precarious [9].
- Sometimes no design exists that alleviate the usability problems described, e.g. because the changes needed conflict with other requirements of the design or dictate extremely complex functionality. Designers may waste resources in trying to cope with such problems.
- Generation of lists of usability problems may not matter much in practical systems development. Wixon [13] comments on a recurring discussion regarding comparison of evaluation techniques that '[i]t is short sighted in that it ignores that problems should be fixed and not just found'.

Taken together, these limitations suggest that it is feasible to examine alternatives or supplements to problem identification and description as the goal underlying the creation and comparison of usability evaluation techniques.

This paper explores if and how redesign proposals may supplement problem descriptions as valuable input from usability evaluation to practical systems development.

2. PROBLEMS AND REDESIGNS
Only few studies have investigated redesign proposals as an outcome of usability evaluation [2,8,11,12]. For example, Dutt et al. [2] considers the ability of heuristic evaluation and cognitive walkthrough to produce requirements for redesigns. While requirements are related to a specific technique, the study doesn't describe the format or nature of those requirements. The study by Sawyer et al. [12] on the impact of inspections on software development suggests that '[p]roviding specific recommendations to fix specific problems has a tremendous positive effect: The development group need not spend time thinking of a solution, plus we gain a psychological advantage in offering constructive suggestions rather than just criticism' (p. 379). This study, however, does not compare usability problems and redesigns, nor point out particularly useful aspects of redesign proposals.

In practical usability work, redesign proposals are often made in the form of quick fixes. Dumas et al. [1] mentions how usability reports from teams of expert evaluators often include proposals for how to fix problems. Usually, however, the quick fixes are only as brief as problem descriptions. They suffer from some of the same limitations that were attributed to usability problems in the introduction. Further, proposals are sometimes quite vague, leading the authors to question 'would the developer who created this site be able to make better choices from these suggestions?' (p. 29). This suggests that some more developed form of redesign proposals could be feasible.

In summary, related work provide some arguments for redesign proposals as (part of) the result of usability evaluation. None of the studies, however, have moved beyond quick fixes integrated with or quite similar to usability problems. Thus, little is known about the utility of redesign proposals, especially of their relative merits compared to problem descriptions.

3. INTERVIEWS WITH DEVELOPERS

As part of a study that compared evaluation techniques we interviewed developers about their perception of usability problems and redesign proposals. Details of the study will be reported elsewhere; here we focus just on interviews with developers.

Forty-three undergraduate and graduate students chose to conduct the evaluation and redesign in a class on HCI and systems design. They evaluated one of Denmark's largest job portals, www.jobindex.dk. The evaluators had one week to conduct the evaluation, and performed it individually. They were told to use approximately eight to ten hours on conducting and reporting the evaluation. Twenty-one evaluators received reference [10] as description of think aloud user testing; twenty-two evaluators received reference [5] as description of the usability inspection technique called metaphors of human thinking.

After completing the evaluation, each evaluator produced three redesigns, one for each of the three parts of Jobindex evaluated. Thirty-six evaluators handed in redesigns, for which they had been asked to use around ten hours. Evaluators were told to create redesigns that addressed some of the usability problems they considered to be the most critical for the users of the application. They were told to imagine that they should provide input for a discussion of whether a redesign decision should be worked out into further detail and possibly be implemented. Evaluators were asked to provide (1) a brief summary of the redesign; (2) a brief argument why the proposed redesign is important; (3) an up to one page explanation of interaction and design decisions in the redesign; and (4) up to two pages of illustrations of how the redesign works.

In practical usability work, the development team has a decisive role in choosing which usability problems to correct and which redesign proposals to follow. Therefore, problems and redesign proposals were assessed by four core members of the development team at Jobindex: (a) the founding director who plays a crucial role in the development; (b) two developers each working on and responsible for parts of the application that were evaluated; (c) a web content manager, responsible for a part of the application evaluated. For brevity, we refer to these four persons as developers. The developers individually assessed a selection of problem descriptions and redesign proposals. Problems and redesigns were presented to developers in a randomized order, alternating between 11 problems, a redesign proposal, 11 problems, etc. One of the developers rated all problems and redesign proposals; the other developers rated those problems and redesigns concerning the part of the application that they work on. The results of the assessment is not included in this paper.

Approximately a week after developers had finished assessing the usability problems and redesign, we conducted individual interviews with them. We asked about their background, experience with rating problems, and impressions of the qualities of redesigns and problems. In addition, we presented them with examples of problems and redesigns that they had assessed as having high or low utility, and asked for their reasons for the assessment. Because the web content manager was working on a part of the application mainly delivering information, we did not interview that developer about redesigns (as this would have regarded changes to content only, not the more complex interaction parts of the user interface). Each interview lasted around an hour.

3.1 Descriptions of usability problems

All developers felt that they already knew most of the problems described by the evaluators. One of the developers said, for example, 'There is not so much new in it' and continues:

> the issues that have been identified, they are either issues which we do not judge as very important, or issues we were well aware of already and with which we knew there were problems … but have not had the time to deal with

While agreeing on the problems, developers appeared to assess severity somewhat differently from evaluators. One of the developers expressed surprise that evaluators had taken such effort to point out a problem he agreed existed but otherwise considered minor. Another said that 'practical experience shows that users can do that', practical experience probably referring to the web logs. Of those usability problems developers said they did not know, actual bugs were given much attention, e.g. 'that [a problem description] is one of our serious problems, it is a bug that we have been chasing without being able to find its cause … such a bug has a high priority on our list'.

The developers' main uses of the problems seemed more to be for prioritizing what to do something about and for confirming design decisions nearing completion, rather than for getting surprising new information. For example,

> usability problems … what one cares about is the extent of them, how many is saying that some thing is a problem and how many is saying that some other thing is a problem, that help me prioritize what I should focus on

An aspect of usability problems emphasized by one of the developers was the reference to users and their problems, e.g. 'I liked best those [problems] that said that the users … that the user tests showed something'.

The developers also noted limitations in the problem descriptions which impacted their utility in the systems development. For example, when seeing a problem again during the interview, one of the developers gave the following example:

> so if an evaluator's comment is that the password is too short, then my comment is: what do you mean by that, too short for what? Exactly because it is short users may be able to remember it, but if he says that the password is too short because a hacker could log in and steal you personal information, then I could say OK now we are talking about that problem

Thus, the lack of clear reasons why something is a problem was considered a shortcoming. Occasionally, problem descriptions would point out something as a problem, but ignore that alternative designs would lead to similar or worse usability problems. In discussing how to show hits of a search in job advertisements, one developer argued:

> ok, so you cannot see where the hit was…on the other hand if we presented the [place in the add] where the hit was instead of the nice form of the add, then that would lead to problems also…so you present a problem, but what is the solution to that problem…sometimes you have, you have some alternatives [to the currently implemented solution], but because there is a problem with one alternative then it is not sure that the other [alternative] is better

Finally, some of the descriptions of usability problems would ignore issues outside of the development team's control. Some problems suggested changing the label of a button for uploading an image to which one of the developers commented that 'we don't have control over the text on it' (because this is done by the operating system) and thus considered that problem to be of low utility.

3.2 Redesign proposals

Compared to usability problems, the single most frequent comment about redesign proposals is that they give good ideas. For example:

> ok, there were some pearls in it … sometimes things that we had not thought about, especially redesign proposals for saying, ok that way of doing it is also possible

And later on remarks that:

> in some situations you may do things one way or the other, and then you can just choose, i.e. whether some list should be alphabetical or just split up…in other situations, like the three level hierarchical selection of job titles, no matter what we do we get into some complicated mess…so if one can find some way of making it more intuitive and usable than other ways, then we accept it eagerly, [because] we haven't quite figured out how to do it ourselves

This input seems especially welcome when developers are tackling a 'particularly hard nut to crack', or when they are just looking for information on 'what is a good idea to get on'.

During all interviews, we asked developers if they could recall usability problems and redesign proposals. Usability problems were mostly remembered by developers as classes of problems, the particular instances was forgotten. One developer said that 'yes, there are several of them [usability problems] that I can still remember' and went on to expand on how redesign proposals on exploring similarities to standard search engines could be incorporated in the design. All developers were, however, able to describe in some detail redesign proposals which they had found interesting:

> for example, someone came with a simple solution to a problem that we have had for a long time: we have a selection box where you may choose counties and cities, which we put into the same selection box … someone suggest why don't you split it up so that you can either select a county or a city or a country … make three lists instead of one … that is one way of doing it which we did not consider previously

A number of attributes of redesigns seem to work well in the developers' opinions. For example, the illustrations (evaluators mostly did these as drawings or mock-ups in HTML) were well liked. For example,

> I think it was those [redesign proposals] that I gave a high assessment, they were really interesting … yes, both of them were characterized by, well they [the evaluators] had grabbed a pencil and made a drawing and said: you could make it in such and such way, thought out of the box so to speak…that is probably the single most positive thing in the entire file [of redesigns and usability problems]

Two developers found the redesigns more concrete than problem descriptions, meaning that they were more clear about what evaluators had in mind when describing the redesign. One of the developers emphasized how, as a form of communication, the redesigns were much more constructive: 'it is almost obvious that it is better to say: if it were this way it was better, rather than just saying: this is wrong… so say this is wrong and here is the alternative'. And finally, all developers stressed how the redesign proposals felt more coherent and complete, i.e. 'there were more meat in them' and 'there is a little more thought in it, a little more completeness'.

As with usability problems, developers pointed out several limitations of the redesigns. For example, some of the redesigns were descriptions of 'more radical proposals for changes, how you can make the things by advanced Java script and stuff like that, that is a new idea but not one that we can use because it is too complicated'. Thus, technical feasibility and coherence with the overall use of technology meant that this proposal did not have much utility for the developer. Similarly, a developer said, reflecting upon a redesign proposal that he recalled: 'then it begins to get confused and complex … and the problem starts to grow … but there are no thoughts on which consequences do this have in the rest of the system'.

Still other redesign proposals were put aside because they did not fit with the printing of resumes on paper that the application were also used for.

Even when redesigns were put aside for reasons like above, developers found them to be of utility. For example, one developer noted that he considered the problem a particular redesign tried to solve to be irrelevant, still the solution was interesting: 'this particular one I can remember because it is the right solution, but the wrong rationale'. Another example is when the proposed solution does not feel right to the developer, but the idea behind the solution is fine, e.g. 'I think that the idea that the user can write and add [job descriptions] is not bad at all, but I am not convinced it should be done in this way'.

3.3 General comments on input from usability evaluation

All developers expressed that both usability problem descriptions and redesign proposals were of very high quality,

e.g. 'they are quite good, both the comments and the redesigns, they capture very well what we are trying to do and come up with some good proposals'. We also asked developers if they would want to receive only problems or redesigns, and all expressed that they wanted to receive both.

Across usability problems and redesign proposals, developers expressed that problems of utility to them were problems that could be fixed easily and quickly. One developer explained:

> typically if something can be easily and quickly fixed … that is a suggestion which requires four months of development is not as useful as some small suggestion, which corrects a small problem in 10 minutes, then I can correct it immediately

In fact, developers and the web content manager all had corrected one or more problems when we interviewed them, approximately one week after having worked through the problems and redesigns.

4. CONCLUSION

The study shows that developers value redesign proposals as input to their development work. The interviews suggest that (a) redesign proposals help developers understand usability problems, i.e. redesigns contribute to characterizing and making more concrete the problems found, and illustrate why problems are important; and (b) redesign proposals are useful for inspiration and for seeking alternative solutions for problems that the development team has been struggling with. These comments do not mean, however, that developers did not appreciate usability problems, especially when they are well argued, clearly described, documented, and easy to fix. On the contrary, all developers wanted both problems and redesign proposals as input from usability evaluation to systems development.

These results suggest that usability evaluations should place more focus on developing and reporting such proposals than is typically done.

The results stand in contrast to the scientific literature on usability evaluation techniques, which largely ignore proposals for redesigns as input to systems development. Redesign proposals may help move beyond Wixon's [13] complaint that most comparisons of usability evaluation techniques focus exclusively on the techniques' ability to generate problems, ignoring what is needed in practical systems development. Moreover, focusing on redesign proposals may help improve the validity of comparisons of usability evaluation techniques, the limitations of which have been pointed out by several authors [3,6]. This could be expected because redesign proposals, according to the developers interviewed, are more concrete, more relevant to their work, and better able to give a clear understanding of what an evaluator intended.

5. REFERENCES

1. Dumas, J., Molich, R., & Jefferies, R. Describing Usability Problems: Are We Sending the Right Message?, *interactions*, 4 (2004), 24-29.
2. Dutt, A., Johnson, H., & Johnson, P. Evaluating Evaluation Methods, *Proc. HCI 1994*, Cambridge University Press (1994), 109-121.
3. Gray, W. D. & Salzman, M. C. Damaged Merchandise? A Review of Experiments That Compare Usability Evaluation Methods, *Human-Computer Interaction*, *13*, 3 (1998), 203-261.
4. Hartson, H. R., Andre, T. S., & Williges, R. C. Criteria for Evaluating Usability Evaluation Methods, *International Journal of Human-Computer Interaction*, *13*, 4 (2001), 373-410.
5. Hornbæk, K. & Frøkjær, E. Evaluating User Interfaces With Metaphors of Human Thinking, *Lecture Notes in Computer Science 2615*, Springer (2002), 486-507
6. Hornbæk, K. & Frøkjær, E. Two Psychology-Based Usability Inspection Techniques Studied in a Diary Experiment , *Proc. NordiCHI 2004*, ACM Press (2004)
7. John, B. & Mashyna, M. M. Evaluating a Multimedia Authoring Tool With Cognitive Walkthrough and Think-Aloud User Studies, *CMU-HCII-95-105 / CMU-CS-95-189* (1995).
8. Johnson H., Generating User Requirements From Discount Usability Evaluations, in Harris, D. *Engineering Psychology and Cognitive Ergonomics, Vol. 2,* Ashgate Publishing, 1997, 339-357.
9. Lavery, D., Cockton, G., & Atkinson, M. P. Comparison of Evaluation Methods Using Structured Usability Problem Reports, *Behaviour and Information Technology*, *16*, 4/5 (1997), 246-266.
10. Molich, Rolf, User testing, Discount user testing, 2003, www.dialogdesign.dk.
11. Muller, M. J. & McClard, A. Validating an Extension to Participatory Heuristic Evaluation: Quality of Work and Quality of Work Life, *Proc. CHI'95*, ACM Press (1995), 115-116.
12. Sawyer, P., Flanders, A., & Wixon, D. Making a Difference - The Impact of Inspections, *Proc. CHI'96*, ACM Press (1996), 376-382.
13. Wixon, D. Evaluating Usability Methods: Why the Current Literature Fails the Practitioner, *interactions*, *10*, 4 (2003), 29-34.

Integrating Usability Design and Evaluation: Training Novice Evaluators in Usability Testing

Mikael B. Skov and Jan Stage
Department of Computer Science
Aalborg University
Aalborg Øst, Denmark
+45 9635 8080
{dubois, jans}@cs.aau.dk

ABSTRACT
This paper reports from an empirical study of training of usability testing skills. 36 teams of novice evaluators with an interest but with no education in information technology were trained in a simple approach to web-site usability testing that can be taught in less than one week. The evaluators were all first-year university students. The paper describes how they applied this approach for planning, conducting, and interpreting a usability evaluation of the same web site.

We discover that basic usability testing skills can be developed. The student teams gained competence in defining good task assignments and ability to express the problems they found. On the other hand, they were less successful when it came to interpretation and analytical skills. They found quite few problems, and they seemed to lack an understanding of the characteristics that makes a problem list applicable.

Keywords
Usability test, training novices, dissemination of usability skills

INTRODUCTION
Despite several years of research on usability testing and engineering, many computer-based information systems still suffer from low usability [4]. One problem arises from the fact that planning and conducting full-scale usability tests yields key challenges of e.g. user integration [7]. Considerable costs arise when a large group of users is involved in a series of tests. Furthermore for some applications it is difficult to recruit prospective test subjects [2].

The theoretical usability evaluation approach denoted as heuristic inspection evolved as a creative attempt to reduce such costs of usability evaluations [5, 6, 8]. The idea in heuristic inspection is that an interface design is evaluated by relating it to a set of guidelines, called heuristics [8]. The aim of the heuristics is to equip people who are not usability specialists to conduct heuristic inspections. Some of the empirical studies of the approach have been based on university students or readers of a computer magazine who act as evaluators [8].The idea behind heuristic inspection is to accomplish a simplified way of conducting usability tests. However, the empirical results indicate that we move the problem from finding users to finding user interface specialists. For a small organization developing web-based systems both of these problems may be equally hard to overcome. On a more general level the relevance of heuristic inspection can also be questioned. It has been argued that real users are an indispensable prerequisite for usability testing. If they are removed, it is at the expense of realism [10].

In this paper, we pursue a different idea of enhancing the knowledge of usability for software designers. One key problem in improving the usability of systems is the challenges involved in the interplay between the design and the evaluation of the system. Sometimes these activities are separated and detached making the interplay difficult and challenging e.g. one potential problem arises from the fact that designers and evaluators do not share a common language or set of tools in order to communicate. Our study explores how we can enhance usability testing competences for novice evaluators. Our aim is to train novice evaluators and compare their usability testing performances against the performances of professional usability testing labs. For our study, we use first-year university students as novice evaluators. First, we outline the taught usability testing approach and present the experiment behind the paper. Secondly, we compare the performances of the novice evaluators to the performances of professional labs on 17 different variables. Finally, we discuss and conclude our study.

METHOD
We have made an empirical study of the usability approach that was taught to the novice evaluators.

Usability Testing Approach
The approach to usability testing was developed through a course that was part of a curriculum for the first year at Aalborg University, Denmark. The overall purpose of the course was to teach and train students in fundamentals of

usability issues and testing. The course included ten class meetings each lasting four hours that was divided between two hours of class lectures and two hours of exercises in smaller teams. All class meetings except for two addressed aspects of usability and testing. The course required no specific skills within information technology that explains the introduction of course number one and five. The purpose of the exercises was to practice selected techniques from the lectures. In the first four class meetings, the exercises made the students conduct small usability pilot tests in order to train and practice their practical skills. The last six exercises were devoted to conducting a more realistic usability test of a specified web site.

The course introduced a number of techniques for usability testing. The first one was the technique known as the think-aloud protocol, which is a technique where test subjects are encouraged to think aloud while solving a set of tasks by means of the system that is tested, cf. [7]. The second technique is based on questionnaires that test subjects fill in after completing each task and after completion of the entire test, cf. [11]. Additional techniques such as interviewing, heuristic inspection, cognitive walkthroughs, etc. were additionally briefly presented to the students.

The tangible product of the usability evaluation should be a usability report that identifies usability problems of the product, system, or web site in question. We proposed to the students that the usability report should consist of 1) an executive summary (1 page), 2) description of the applied methodology (2 pages), 3) results of the evaluation (5-6 pages), and 4) a discussion of the applied methodology (1 page). Thus, the report would typically integrate around 10 pages of text. It was further emphasized that the problems identified should be categorized, at least in terms of major and minor usability problems. In addition, the report should include all data material collected such as log-files, tasks for test subjects, questionnaires etc.

Web-Site

Hotmail.com was chosen as object for our study mainly for two reasons. First, hotmail.com is one of the web-sites that provides advanced features and functionalities appropriate for an extensive usability test. Furthermore, hotmail.com facilitates evaluations with both novice and expert test subjects due to its vast popularity. Secondly, hotmail.com has been of focus in other usability evaluations and we compare the results of the student teams in our study with other results on usability evaluations of hotmail.com (further explained under Data Analysis).

Subjects

The subjects were all first-year university students enrolled in four different studies at the faculty for natural sciences and engineering at Aalborg University; the four studies were architecture and design, informatics, planning and environment, and chartered surveyor. None of the subjects indicated any experiences with usability tests prior to the study.

36 teams involving a total of 234 subjects (87 females, 37%) participated in our study of which 129 (55%) acted as test subjects, 69 (30%) acted as loggers, and 36 (15%) acted as test monitors, cf. [10]. The average subject age was 21.2 years old (SD=1.58) and the average team size was 6.5 subjects (SD=0.91). The average size of number of test subject in the teams was 3.6 subjects (SD=0.65). 42 (33%) of the 129 test subjects had never used hotmail.com before the conduction of test, whereas the remaining 86 subjects had rather varied experience.

Procedure

The student teams were required to apply the techniques presented in the course. Additionally, each team was required to select among themselves the roles of test subjects, loggers, and test monitor.

The test monitor and the loggers received after the second lecture a two-page scenario specifying the web-based mail service www.hotmail.com as the object of focus in the test. The scenario also specified a comprehensive list of features that emphasized the specific parts of www.hotmail.com they were supposed to test. The test monitor and the loggers would then start to examine the system, design tasks, and prepare the test in general, cf. [10]. The www.hotmail.com web site in the study was kept secret to test subjects until the actual test conduction.

30 (83%) of the 36 teams provided information on task completion times for 107 (83%) of the 129 subjects resulting in an average session time of 38.10 minutes (SD=15.32 minutes). Due to the pedagogical approach of the university, each team was allocated their own offices equipped with a personal computer and Internet access. Most teams conducted the tests in these offices. After the tests, the entire team worked together on the analysis and identification of usability problems and produced the usability report.

Data Analysis

The 36 usability reports were the primary source of data for our empirical study. The 36 reports had an average size of 11.36 pages (SD=2.76) excluding the appendences, which had an average size of 9.14 pages (SD=5.02). All reports were analyzed, evaluated, and marked by both authors of this paper according to the following three steps.

1) We designed a scheme for the evaluation of the 36 reports by analyzing and evaluating five randomly selected reports from the 36 reports. Through discussions and negotiations we came up with an evaluation scheme with 17 variables as illustrated in table 3. The 17 variables was divided into three overall categories of evaluation (relates the conduction of the test), report (relates the presentation of the test and the results), and results (relates the results and outcome of the usability test). Finally, we described, defined, and illustrated all 17 variables in a two-page marking guide.

2) We worked individually and marked each report in terms of the 17 variables using the two-page marking guide. The

Team	Conduction			Documentation					
	Test procedure conduction	Task quality and relevance	Question-naire / Interviews	Test description	Data quality	Clarity of problem list	Executive summary	Clarity of report	Layout of report
Student (N=36)	3.42 (0.73)	3.22 (1.05)	2.72 (1.00)	3.03 (0.94)	3.19 (1.33)	2.53 (1.00)	2.39 (0.80)	2.97 (0.84)	2.94 (0.89)
Professional (N=8)	**4.38 (0.74)**	3.13 (1.64)	3.50 (1.69)	**4.00 (1.31)**	2.13 (0.83)	**3.50 (0.93)**	**3.38 (1.06)**	**4.25 (0.71)**	3.25 (0.71)

Team	Results							
	Number of problems*	Problem categorization	Practical relevance	Qualitative results overview	Quantitative results overview	Use of literature	Conclusion	Evaluation of test
Student (N=36)	2.56 (0.84)	2.06 (1.22)	3.03 (1.00)	3.03 (1.00)	2.28 (1.14)	3.08 (0.81)	2.64 (0.90)	2.44 (1.08)
Professional (N=8)	**4.13 (1.13)**	3.25 (1.75)	**4.25 (1.49)**	**3.75 (1.16)**	2.00 (1.51)	3.13 (0.35)	**3.88 (0.64)**	2.88 (1.13)

Table 3. Mean values and standard deviation (in parentheses) of all 17 variables for the student and professional teams. The grade for the number of identified problems is calculated from the actual number of identified usability problems in each usability report according to the following procedure: 1 = 0-3 identified problems; 2 = 4-7 identified problems; 3 = 8-12 identified problems; 4 = 13-17 identified problems; 5 = >17 identified problems. Boldfaced numbers indicate significant differences between the student and professional teams.

markings were made on a scale of 1 to 5 (1=no or wrong answer, 2=poor or imprecise answer, 3=average answer, 4=good answer, and 5=outstanding answer). We furthermore counted the number of identified usability problems in all 36 usability reports. In our study, we define a usability problem as the prevention or impediment of realization of user objectives through the interface. Furthermore, we specified limits for grading afterwards based on their distribution on the scale (1=0-3 problems, 2=4-7 problems, 3=8-12 problems, 4=12-17 problems, and 5>17 problems).

3) All reports and evaluations were compared and a final evaluation on each variable was negotiated. In case of disagreements on marking, we pursued the following two-folded procedure - 1) if the difference was equal to one grade we would renegotiate the grade based upon our textual notes 2) if the difference was equal to two grades, we would reread and reevaluate the report in a collaborative effort focusing only on the corresponding variable. For our study, no disagreement exceeded more than two grades.

To examine the overall performance of the students, we included two additional sets of data in the study. First, we compared the student reports to usability reports produced by teams from professional laboratories. These reports were selected from a pool of usability reports produced in another research study where nine different usability laboratories received the same scenario as outlined above and conducted similar usability tests of www.hotmail.com, cf. [2]. Of these nine usability reports, we dropped one due to its application of only theoretical usability evaluation techniques, e.g. heuristic inspection, thereby not explicitly dealing with the focus of our study namely user-based testing techniques. The remaining eight usability reports were analyzed, evaluated, and marked through the same procedure as the student reports. We analyze the data using Mann-Whitney U Test for testing the significance between means for all 17 variables

RESULTS

The general impression of the results as outlined in table 3 suggests that the professional laboratories performed better than the student teams on most variables. However, on three, e.g. 2), 5), 14), of the 17 variables the student teams actually performed best, whereas on the remaining 14 variables the professional teams on average did better and for six variables, e.g. 1), 8), 10), 11), 12), 16, the professional teams were marked one grade (or more) higher than the students.

Conducting and Documenting the Usability Test

Test conduction relates the actual conduction of the usability test. The professional teams have average of 4.38 (SD=0.74) almost one grade higher than the student teams and a Mann-Whitney U Test shows strong significant difference between test conduction of the student teams and test conduction of the professional teams ($z=-2.68$, $p=0.0074$). On the other hand, even though the students performed slightly better on the quality and relevance of tasks, this difference is not significant ($z=0.02$, $p=.984$). Finally, no significant variation was found for the questionnaires and interview guidelines quality and relevance ($z=-1.63$, $p=0.1031$).

Concerning presentation of the usability testing results, the professional teams did better than the student teams on clarity of the usability problem list and we found strong

significant variance on this variable (z=-2.98, p=0.0029) and we also found strong significant difference on the clarity of the entire report (z=-3.15, p=0.0016). Further, there is significant difference on the teams' description of the test (z=-2.15, p=0.0316) and on the executive summary (z=-2.27, p=0.0232). The student teams actually performed significantly better than the professional teams on the quality of the data material in the appendix (z=2.07, p=0.0385). Finally, no significance was identified for the layout of the report (z=-1.02, p=0.3077).

Identification and Categorization of Test Results

The pivotal results of all student and professional usability reports were the identification (and categorization) of various usability problems. However, the student and professional teams performed rather differently on this issue. The student teams were on average able to identify 7.9 usability problems (in the marking scale: Mean 2.50, SD 0.88) whereas the professional teams on average identified 21.0 usability problems (in the marking scale: Mean 4.13, SD 1.13) and a Mann-Whitney U Test confirms strong significance (z=-3.09, p=0.002). However, the professional teams actually performed rather dissimilar identifying from seven to 44 usability problems.

The student teams provided better overview of the quantitative results, but this difference was not significant (z=0.90, p=0.3681). On the hand, the practical relevance of the identified usability problems was significantly higher for the professional teams (z=-2.56, p=0.0105). Furthermore, the conclusion are better in the professional team reports and this difference was strong significant (z=-3.13, p=0.0017). The overview of the qualitative results also showed significant variance (z=-1.99, p=0.0466). No significance was found for the problem categorization (z=-1.84, p=0.0658), the use of literature (z=-0.05, p=0.9601), or the evaluations of the test procedure (z=-1.00, p=0.3173).

DISCUSSION

Our aim with this study was to explore dissemination of usability testing skills to people with no formal training in information technology design or use. Previous studies have suggested heuristic inspection as a creative attempt to reduce costs of usability evaluations. Research has shown that planning and conducting full-scale usability tests yields key challenges of e.g. user integration [7]. Considerable costs arise when a large group of users is involved in a series of tests. Further, for some applications it is difficult to acquire prospective test subjects [2]. However, user-based evaluations may provide more valid results.

Our study documents experiences from a course with 234 students that conducted a usability test of hotmail.com in teams of four to eight students. The results of these tests were documented in 36 individual usability reports. Our study reveals a number of interesting issues to consider when novices are to conduct full-scale user-based usability evaluations.

One key finding of our study is characteristics of usability problem identification (and categorization). The student teams are only able to identify significantly fewer problems than the professional teams. A key aim in usability testing is to uncover and identify usability problems, and the student teams on average found 7.9 usability problems whereas the professional teams on average found 21 usability problems. The student teams perform rather differently on this variable as one team identify no problems (it seems this team misunderstood the assignment) to two teams identifying 16 problems. Most of the teams identify no more than 10 problems. The professional teams also perform rather differently and this is perhaps more surprising where one team identify 44 problems and one team identify only seven problems. The latter is actually rather disappointing for a professional laboratory. We are in process of analyzing the severity of the problems and we do not have any results on this issue so far.

Related the conduction of the usability test sessions, the majority of student teams score 4, which indicates well-conducted tests with a couple of problematic characteristics. The average on 3.43 also reflects the general quality of the test processes. The professional laboratories score an average of 4.6 on this factor, and 6 out of 8 score the top mark. This is as it should be expected because experience will tend to raise this variable. However, the student teams perform rather well with respect to planning and conducting the usability testing sessions. On the other hand, there seems to be no direct correlation between the quality of the test conduction or the quality of the assigned tasks and the number of identified problems. Thus, the students may plan their evaluations carefully, but

Another variable that exhibits a difference is the practical relevance of the problem list, cf. figure 5. The student teams are almost evenly distributed on the five marks of the scale, and their average is 3.2. Yet when we compare these to the professional laboratories, there is a clear difference. The professionals score an average of 4.6 where 6 out of 8 laboratories score the top mark. This difference can partly be explained from the experience of the professionals in expressing problems in a way that make them relevant to their customers. Another source may be that the course has focused too little on discussing the nature of a problem; it has not been treated specifically with examples of relevant and irrelevant problems.

Our study is limited in a number of different ways. First, the environment in which the tests were conducted was in many cases not optimal for a usability test session. In some cases, the students were faced with slow Internet access that influenced the results. Second, motivation and stress factors could prove important in this study. None of the teams volunteered for the course (and the study) and none of them received any payment or other kind of compensation; all teams participated in the course because

it was a mandatory part of their curriculum. This implies that students did not have the same kinds of incentives for conducting the usability test sessions as people in a professional usability laboratory. Thirdly, the demographics of the test subjects are not varied with respect to age and education. Most test subjects were a female or a male of approximately 21 years of age with approximately the same school background and recently started on a design-oriented education. The main difference is the different curricula they follow. Fourthly, the hotmail.com website is a general website in the sense it provides no or little domain knowledge. Different distributions on the variable may emerge for more specialized user interfaces, see [2] for examples.

CONCLUSION

The existing low level of skills in usability engineering among web-site development teams is likely to prohibit moves towards the ideal of universal access and the idea of anyone, anywhere, anytime. This article has described a simple approach to usability testing that aims at quickly teaching fundamental usability skills to people without any formal education in software development and usability engineering. Whether this approach is practical has been explored through a large empirical study where 36 student teams have learned and applied the approach.

The student teams gained competence in two important areas. They were able to define good tasks for the test subjects, and they were able to express the problems they found in a clear and straightforward manner. Overall, this reflects competence in planning and writing. The students were less successful when it came to the identification of problems, which is the main purpose of a usability test. Most of the teams found too few problems. It was also difficult for them to express the problems found in a manner that would be relevant to a practicing software developer.

The idea of this approach is to reduce the efforts needed to conduct usability testing. This is consistent with the ideas behind heuristic inspection and other walkthrough techniques. On a more general level, it would be interesting to identify other potential areas for reducing effort.

This approach to usability testing did provide the students with fundamental skills in usability engineering. Thus it is possible to have usability work conducted by people with primary occupations and competencies that are far away from software development and usability engineering. We see the approach as a valuable contribution to the necessary development emphasized here: "Organizations and individuals stuck in the hierarchies and rigidity of the past will not foster what it takes to be successful in the age of creativity, the age of the user, and the age of the Internet economy" [1].

ACKNOWLEDGMENTS

We would like to thank the participating students in the study. In addition, we would like to thank the anonymous reviewers for comments for earlier drafts.

REFERENCES

1. Anderson, R. I. Making an E-Business Conceptualization and Design Process More "User"-Centered. *interactions 7*, 4 (July-August), 27-30.
2. Kjeldskov, J. and Skov, M. B. (2003) Evaluating the Usability of a Mobile Collaborative System: Exploring Two Different Laboratory Approaches. In Proceedings of the 4th International Symposium on Collaborative Technologies and Systems, pp. 134 - 141
3. Kjeldskov, J. and Skov, M. B. (2003) Creating Realistic Laboratory Settings: Comparative Studies of Three Think-Aloud Usability Evaluations of a Mobile System. In Proceedings of the 9th IFIP TC13 International Conference on Human-Computer Interaction (Interact2003), IOS Press, pp. 663 – 670.
4. Molich, R. *Comparative Usability Evaluation Reports*. Available at http://www.dialogdesign.dk/cue.html.
5. Molich, R., and Nielsen, J. Improving a Human-Computer Dialogue. *Comm. ACM 33*, 3, 338-348.
6. Nielsen, J. Finding Usability Problems Through Heuristic Evaluation. In *Proceedings of CHI '92*, ACM Press, 373-380.
7. Nielsen, J. *Usability Engineering*. Morgan Kaufmann Publishers, 1993.
8. Nielsen, J., and Molich, R. Heuristic Evaluation of User Interfaces. In *Proceedings of CHI '90*, ACM Press, 249-256.
9. Rohn, J. A. The Usability Engineering Laboratories at Sun Microsystems. *Behaviour & Information Technology 13*, 1-2, 25-35.
10. Rubin, J. *Handbook of Usability Testing. How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, 1994.
11. Spool, J. M., Scanlon, T., Schroeder, W., Snyder, C., and DeAngelo, T. *Web Site Usability. A Designer's Guide*. Morgan Kaufmann Publishers, 1999.
12. Sullivan, T., and Matson, R. Barriers to Use: Usability and Content Accessibility on the Web's Most Popular Sites. In *Proceedings of Conference on Universal Usability* (Washington, November 2000), ACM Press, 139-144

A Multi-Perspective Approach to Tracking the Effectiveness of User Tests: A Case Study

Effie Lai-Chong Law
Computer Engineering and Networks Laboratory (TIK)
Swiss Federal Institute of Technology (ETH Zürich)
Gloriastrasse 35, CH-8092 Zürich, Switzerland
law@tik.ee.ethz.ch

ABSTRACT
In this paper we delineate a multi-perspective approach to tracking the effectiveness of user tests, which have been performed on a web-based educational system. We have identified a definitional issue about the effectiveness of usability evaluation methods and thus proposed that tracking and supporting the integration of usability evaluation results should be an integral part of usability engineering process. We also identified a theoretical void in studying the persuasive power of usability evaluation results and thus proposed to bridge the gap with process theories of persuasion. We have collected data from several sources representing different roles and perspectives – usability practitioner, system developer, system manager, and representative end-users. We have consolidated the multi-perspective data to address several hypotheses that predict the persuasiveness of different qualities of usability problems to induce fixes and the effectiveness of such fixes. Implications for future research on this specific topic are inferred.

Categories and Subject Descriptors
H.5.2. Information Interfaces and Presentation: User Interface – Evaluation/methodology

General Terms
Experimentation, Measurement

Keywords
Effectiveness, User Test, Usability Problem, Persuasion Theories

1. INTRODUCTION

Effectiveness is a key notion in usability research. Nevertheless, up to now there has not yet been any well defined parameter that can be used as a reliable and valid indicator of the effectiveness of usability evaluation methods (UEMs). Given the ultimate goal of usability evaluation is to improve the system of interest, the evaluation tool or method selected can be proved to be effective only if such a goal can be attained, ideally at an optimal cost. A number of studies on comparing the effectiveness of different UEMs were conducted in early 1990s [e.g., 5, 10, 13, 19]. Unfortunately, these studies were harshly criticized as lack of experimental rigor and the outcomes were regarded as dubious [8]. In some recent related works [e.g., 4, 9, 15, 22], two parameters – thoroughness and validity – have been adopted to define the effectiveness of UEMs. However, such a definition seems oversimplified with the primary goal of usability evaluation of improving a system remaining unfulfilled. Usability evaluation should not cease at the point when a list of UPs is produced [23]. More important is to insure that such a list can somehow render the system more usable and useful. Indeed, usability evaluation results can have stronger impacts when developers are provided concrete and feasible improvement suggestions from users and usability practitioners than when they are merely confronted with negative criticisms. We advocate that tracking the effectiveness and supporting the incorporation of usability evaluation results into the improvement of the system tested should be an integral part of the overall usability engineering process.

Further, we observe that there is a theoretical void in studying the persuasive power of usability evaluation results and attempt to bridge the gap with process theories of persuasion. Based on the assumption that the effectiveness of a UEM is a combinatorial parameter, we employ multiple methods to collect data from different stakeholders involved in the usability evaluation. The data thus collected may shed some light onto the significant practical issue about the reliability of user tests, which are normally employed to benchmark other UEMs. In the ensuing text, we will elaborate on the aforementioned theoretical and empirical issues.

2. LITERATURE REVIEW

The literature on tracking the effectiveness of UEMs is actually limited. John and Mark's [12] exploratory work is representative in this area, though its methodology has been challenged [3, 11]. Nonetheless, the case study documented by the two authors illustrates clearly how intricate and resource-demanding such a task can be. According to their model, it is necessary to estimate the values of three key variables: (i) How many of usability problems predicted by a UEM can really be experienced by end-users (*predictive power*)? How many of these usability problems can result in fixes or changes of code (*persuasive power*)? How many of these fixes can really improve the usability of the system (*design-change effectiveness*)? The five UEMs investigated were all predictive or analytic. Respective lists of UPs thus derived were benchmarked with clusters of user tests. John and Mark tracked those UPs predicted by the UEMs of interest and verified by the user tests, but not those UPs that were directly discovered by the user tests and overlooked (i.e. misses) by any of the UEMs.

A method known as RITE (Rapid Iterative Testing and Evaluation) for evaluating the efficacy of fixes of UPs identified in user tests has recently been advocated by Medlock and his colleagues [18, 23]. The key to the success of RITE is the intense participation of at least one member of the development team and he usability engineer. They must communicate seamlessly so that

corrigible UPs can get fixed without having to go through any formal process. Despite its claimed advantages, the generalizability of the RITE method is yet to be demonstrated.

Intuitively speaking, developers and manager are more likely to commit to resolving UPs if they are persuaded about the necessity and utility of potential fixes. This concept of persuasiveness, however, is not well addressed in the literature. In fact, John and Marks [12] do not root their notion of 'persuasive power' in any social cognitive theories. We believe that process theories of persuasion [7], especially their emphasis on distinctive cognitive mechanisms, can shed some light into the issue pertinent to the acceptance and adoption of usability evaluation results by developers, designers, and managers alike. According to N.H. Anderson's information integration theory [2], four general determinants of weight of information are its relevance, salience, reliability and quantity. The heavier the weight, the higher the likelihood the information will be accepted and yield the action (implicitly or explicitly) suggested. Furthermore, McGuire [17] addresses that distal persuasion variables such as recipient's intelligence, motivation and personality; sender's perceived domain-specific expertise; message's fear arousal and communication modality can have effect upon the reception and acceptance of the message content. We infer some implications from these theories to the understanding of usability problems management.

3. RESEARCH HYPOTHESES

We formulate different hypotheses based on the literature perused and summarized above. Four of them are presented here and the others will be reported elsewhere. Put briefly, we assume that the persuasive power of user-test results to induce fixes depends on their saliency and ability to motivate developers (i.e. criticality of UPs) and reliability (i.e., frequency of UPs).

H1a: Due to the saliency effect, UPs rated with higher severity level are more persuasive to induce fixes than those rated with lower severity.

H1b: Fixes of severe UPs are more effective than those of moderate or minor UPs, because developers are more motivated to fix the former than the latter.

H2a: Due to the reliability effect, UPs identified with higher frequency are more persuasive to induce fixes than those with lower frequency?

H2b: Fixes of frequent UPs are more effective than those of rarer UPs, because developers can have more information about the former than the latter.

4. BASELINE MEASUREMENTS

The system on which we performed international user tests (IUT) was a platform (version 0.85; March 2003) designed for enabling the exchange of online educational content among academic and industrial institutions. The interface of this brokerage platform was usability tested with 19 representative end users from four different European countries. Standard user test procedures were adopted [6] and implemented locally with the respective language versions by Local Testers. Each participant was asked to perform ten task scenarios covering the core functionalities of the platform and to *think aloud* to maintain a running commentary as he or she interacted with the system. In the usability evaluation report, for every UP, descriptions (where, what and how), severity level (severe, moderate or minor [1]) and frequency (number of users experienced) were presented.

5. TRACKING EFFECTIVENESS

We tracked the effectiveness of the IUT with the baseline list of 81 UPs, of which 52 were given plausible causes and/or potential redesign solutions by the usability specialist. Specifically, we aim to answer three major questions:

i. How many of the UPs reported have induced fixes?
ii. How persuasive were different UP qualities to induce fixes?
iii. How effective were the fixes?

Three sources of data were collected through different procedures.

5.1 Usability Specialist Review

The usability specialist, who was involved in extracting UPs from the qualitative data of the IUT, re-evaluated each of the 81 UPs observed in the previous version (v. 0.85) with the recent version of the platform (version 1.0; January 2004). She identified those UPs that did not receive any fix and described how the other UPs were fixed.

5.2 Development Team Portfolio

The chief developer and the platform manager, who was heavily involved in deciding which and how UPs to be fixed, were asked to provide data on:

(i) the effort invested or would be invested in fixing the UPs
(ii) the decision-making factors for fixing or not fixing the UPs
(iii) the techniques and references used for implementing the fixes.

The developer described the effort with a five-point scale (very short, short, medium, long, very long). He added brief remarks for 15 UPs with most of them being related to the techniques employed for the actual or would-be changes. The platform manager also added some brief remarks of various natures for 41 UPs.

5.3 End User Retest

Three male participants, who took part in the IUT one year ago, were re-invited to evaluate the current version of the platform. They were all university faculty members with high level of competence in information technology and high level of knowledge about e-Learning (i.e. the domain of the platform evaluated). Their participations were voluntary. In the testing session, they were required to perform a set of 12 task scenarios with nine of them being more or less the same as those they performed in the IUT, and the procedure used was also similar.

6. RESULTS

6.1 Usability Specialist Review

31 out of the 81 UPs identified in the IUT were fixed by the developers. In other words, 50 UPs did not receive any fix. The

Impact Ratio (see Equation 1) is only 38.3%, which is relatively low. We further broke the results down in terms of severity level (Table 1) and frequency (Table 2).

Equation 1 [21]:

$$\text{Impact Ratio (IR)} = \frac{\text{Number of Problems Receiving a Fix}}{\text{Total Number of Problems Found}} * 100$$

Table 1. Impact ratios by problem severity levels

	Minor	Moderate	Severe
With Fix / Change (C)	6	17	8
No Fix / Change (NC)	17	24	9
Impact Ratio (IR)	26.1%	41.5%	47.1%

The IR of severe UPs is higher than that of the other two. However, Chi-Square tests show that there are no significant differences between the cells in Table 1.

Table 2. Impact ratios by problem frequency levels

	Low	Medium	High
With Fix / Change (C)	14	8	9
No Fix / Change (NC)	24	16	10
Impact Ratio (IR)	36.8%	33.3%	47.4%

*Low = single user; Medium = >1 and <=20% of the users; High =>20%

The IR of "High"-UPs is larger than that of the other two, but Chi-Square tests show that there are no significant differences between the cells in Table 2.

6.2 Development Team Portfolio

For each of the 29 out of 31 fixes, the chief developer reported the effort required with the five-point scale mentioned earlier (NB: the detailed results will be reported elsewhere). None of the UPs falls in the category 'very long'. It implies that the developer did not tend to fix any UP entailing much effort.

6.3. End User Retest

The three test participants - P1, P2 and P3 – evaluated the earlier version of the platform about one year ago. The rationales for recruiting "old" participants were to observe whether the UPs they experienced previously would perish or persist and to minimize the user effect [14]. Three separate lists of usability problems were extracted and they were compared with their counterparts obtained in the earlier IUT (Table 3). Note that those UPs associated with the completely new functionalities of the platform to which the three participants had never exposed in the IUT were *not* counted.

Table 3. Main results of end-user retest

	P1	P2	P3
No. of UPs **already** experienced in the **earlier** version	14	26	16
No. of UPs **persistently** experienced in the **current** version	4	3	2
No. of UPs **no longer** experienced in the **current** version	10	23	14
No. of UPs **newly** experienced in the **current** version	5	8	6

An inherent limitation of our study is that the effectiveness of the fixes can only be tracked based on the three users' evaluations. Clearly, the validity and reliability of the results could be higher if more users were involved. Nevertheless, a UP could be experienced by none, one, two or all of the three users in v.0.85, the same UP could also be experienced by none, one, two or all of the three users in v.1.0. We developed a data analysis scheme accordingly (details will be reported elsewhere).

Out of the 31 fixes, 15 were effective or mildly effective, 11 had no effect, four were bad and one was terrible. The reported effort for this terrible fix was "very short". The UP concerned was that the error message was not conspicuous enough to be spotted effectively and its severity level was moderate. The fix involved enlarging the font of the text with the colour remaining the same. The system manager remarked that the fix was based on a 'typical approach' for attracting attention to a message. Two of the five effective fixes involved a relatively high effort (i.e., "long") and both were rated severe, whereas the reported efforts of the other three less severe UPs were "very short" or "short". Moreover, we computed the effectiveness of fixes of UPs of different severity and frequency levels (see Table 4 and Table 5).

Table 4. Fix-effectiveness ratio by severity level

	Severe	Moderat	Minor
Effective# Fixes	5	7	3
Ineffective* Fixes	3	10	3
Fix- Effectiveness Ratio (FER)	38.5%	29.2%	50%

Note: # include mildly effective; * include no effect, bad and terrible fixes

The FER of minor UPs is higher than that of the other two. However, Chi-Square tests show that there are no significant differences between the cells in Table 4.

Table 5. Fix-effectiveness ratio by frequency level

	High	Medium	Low
Effective# Fixes	6	2	7
Ineffective* fixes	3	6	7
Fix-Effectiveness ratio (FER)	66.7%	25%	50%

The FER of "Low"-UPs is larger than that of the other two. However, Chi-Square tests show that there are no significant differences between the cells in Table 5.

7. GENERAL DISCUSSION

In the ensuing text, we will go through the research hypotheses delineated in Section 3. Note that the current work was an exploratory case study aiming to give directions of the related future research. As there was no *a prior* stringent experimental manipulation or control, the results obtained cannot lead to any conclusive claims.

H1a: Severe UPs would be more likely to induce fixes

H1a was not supported statistically. However, results show that the UPs rated with high severity tended to be more persuasive to induce fixes than their less severe counterparts (cf. Impact Ratios in Table 2). Arguing along the line of N.H. Anderson's

information integration theory [2], the weight of a piece of information increases with its saliency and is more likely to capture a recipient's attention. Apparently, a UP tagged with a 'severe' label tends to be more salient than one tagged with a 'minor' label. The heightened saliency and the associated emotional responses (i.e., anxiety or fear) can become a force to drive corrective actions. This mechanism may explain why the severe UPs had a higher rate of receiving fixes.

H1b: Severe UPs would have more effective fixes
H1b was rejected. Fixes of minor UPs tended to be more effective than their more severe counterparts (cf. Fix-Effectiveness Ratios in Table 4), though statistically the difference was insignificant. As minor UPs were generally less complicated than severe UPs, therefore the Fix-Effectiveness Ratio tended to be higher.

H2a: Frequent UPs would be more likely to induce fixes.
H2a was not supported statistically. However, results show that the UPs rated with higher frequency tended to be more persuasive to induce fixes than their less frequent counterparts (cf. Impact Ratios in Table 2). We can again apply the information weight model to explain the observed difference in the tendency to fix. Clearly, it is more convincing that a UP is a real problem if more than one user has experienced it. Indeed, some usability researchers and practitioners tend to discard UPs with single occurrence from further analyses [16], based on the assumption that the peculiarity of users' beliefs and attitudes may play in role in ringing "false alarms".

H2b: Frequent UPs would have more effective fixes
H2b was not supported statistically. However, fixes of highly frequent UPs tended to be more effective than their less frequent counterparts (cf. Fix-Effectiveness Ratios in Table 5). Presumably, the higher the number of users experience a UP, the more elaborated the description of the UP will be, especially the contextual data (cf. Anderson's "relevance"), from which the developer can gain more insights into devising appropriate fixes. This assumption on elaborative-ness (cf. Anderson's "quantity") can somewhat explain the observed difference in the effectiveness of fixes for UPs with different frequencies.

In summary, the results presented above reveal two intriguing facts: First, the outcomes of user tests cannot be effectively incorporated into redesign of a system, considering only 38% of the UPs reported receiving a fix and about 68% (= 15/22) of these fixes were effective or mildly effective (NB: this percentage will be inflated if we take the nine UPs that none of the three users experienced in either of the two versions into account). In other words, approximately only 26% (= 38%*68%) of the results of a user test were applicable in improving the system in question. Second, users could be highly adaptive to the "imperfections" of the system, considering that on average 82.4% (Table 3) of the previously experienced UPs was no longer a nuisance and that 38% (=19/50) of the non-fixed UPs did not cause any further trouble, at least for the three users. Such "self-dissolution" of usability problems can be attributed to different possible reasons: the learnability of the system, the increased tolerance of the user towards design flaws, the giving up of lodging complaints that make no effect (i.e. non-fixed UPs reported in the earlier user test), the overcoming of initial psychological barriers of deploying a new system, etc.

8. CONCLUSION

The current exploratory study is not meant to provide any definitive answers to the issues related to tracking the effectiveness of user tests. Instead, it aims to draw the HCI community to this neglected issue. As demonstrated in the foregoing descriptions, tracking the effectiveness of a user test is very resource-demanding and complex. It is likely to be one of the reasons why usability practitioners do not bother to poke into this question. By the same token, managers do not bother to analyse the ROI (Return On Investment) of usability evaluation [20].

Furthermore, the open problem addressed in the beginning of the paper still remains unanswered: *What is the reliable and valid indicator of the effectiveness of UEM*? While we strongly believe that it should be more than conventionally defined "thoroughness" and "validity", we have not yet been able to derive a neat and tidy mathematical formula, which can reduce a cluster of variables into a single comprehensible and computational entity. Nevertheless, as mentioned above, we posit that the effectiveness of a UEM should be specified with two major terms – *Persuasiveness of Problem*- how many percent of UPs identified can induce a fix and *Efficacy of Fix* - How many of the fixes are effective in the sense that they do not entail any re-fix. Besides, process theories of persuasion [7] should further be explored to study the topic of tracking effectiveness of UEMs.

9. REFERENCES

[1] Artim. J. M. (2003). Usability problem severity ratings. Access at: http://www.primaryview.org/CommonDefinitions/

[2] Anderson, N. H. (1981). *Foundations of information integration theory*. Academic Press.

[3] Carroll, J. M. (1998). On an experimental evaluation of claim analysis. *Behaviour & Information Technology, 17*(4), 242-243.

[4] Cockton, G., & Woolrych, A. (2001). Understanding inspection methods. In A. Blandford, J. Vanderdonckt, & P.D. Gray (Eds.), People and Computer XV (pp. 171-192). Springer-Verlag.

[5] Desurvire, H.W., Kondziela, J.M., & Atwood, M.E. (1992). What is gained and lost when using evaluation methods other than empirical testing. In *Proceedings of CHI'92*.

[6] Dumas, J.S., & Redish, J.C. (1999). *A practical guide to usability testing* (rev. ed.). Exeter: Intellect.

[7] Eagly, A., & Chaiken, S. (1993). *Psychology of Attitudes.* NY: Harcourt, Brace Jovanovic.

[8] Gray, W.D., & Salzman, M.C. Damaged merchandise? *Human-Computer Interaction, 13* (1998), 203-262.

[9] Hartson, H.R., Andre, T.S., & Williges, R.C. (2001). Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction, 13*(4), 373-410.

[10] Jeffries, R., Miller, J.R., Wharton, C., Uyeda, K.M. (1991). User interface evaluation in the real world: A comparison of four techniques. In *Proceedings of CHI'91*.

[11] John, B. (1998). On our case study of claims analysis and other usability evaluation methods. *Behaviour and Information Technology, 17*(4), 244-246.
[12] John. B., & Marks, S.J. (1997). Tracking the effectiveness of usability evaluation method. *Behaviour and Information Technology, 16*(4/5), 188-202.
[13] Karat, C.-M., Campbell, R., & Fiegel, T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. In *Proceedings CHI'92*.
[14] Law, E. L.-C., & Hvannberg, E.T. (2004). Analysis of the combinatorial user effect in international usability tests. In *Proceedings of CHI'04*, April 2004, Vienna, Austria.
[15] Law, E. L-C., & Hvannberg, E. T. (2004). Analysis of strategies for estimating and improving the effectiveness of heuristic evaluation. In *Proceedings of NordiCHI 2004*, 23-27 October, Tampere, Finland.
[16] Lewis, J.R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors, 36*(2), 368-378.
[17] McGuire, W. J. (1968). Personality and Attitude Change: An Information Processing Theory. In Greenwald and Brock (eds.), *Psychological Foundations of Attitude*.
[18] Medlock, C. M., Wixon, D., Terrano, M., Romero, R.L., & Fulton, B. (2002). Using the RITE method to improve products; a definition and a case study. In *Proceedings of UPA'02*.
[19] Nielsen, J., & Philips, V.L. (1993). Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared. In *Proceedings of INTERACT'93*.
[20] Rosenberg, D. (2004). The myths of usability ROI. *Interactions, Sept-Oct*, 23-29.
[21] Sawyer, P., Flanders, A., & Wixon, D. (1996). Making a difference – the impact of inspections. In *Proceedings of CHI'96*.
[22] Sears, A. (1997) Heuristic walkthroughs. *Journal of Human-Computer Interaction, 9*, 3, 213-234.
[23] Wixon, D. (2003). Evaluating usability methods: Why the current literature fails the practitioner. *Interactions, 10, 4*, 29-34.

Cause and Effect in User Interface Development

Ebba Thóra Hvannberg
University of Iceland
Hjardarhaga 2-6
+354 525 4702
ebba@hi.is

ABSTRACT

There is a lack of means of translating or relating work products from elicitation, such as work models, to design and using results of evaluation as feedback to design. This paper suggests that a richer model of evaluation be created that is built concurrently with the design activity and that records the cause / effect relationship between design and the problem domain and the implications work models have on design. It also suggests that the distinction between elicitation and evaluation be diminished. The paper presents two case studies from air traffic control and poses questions that are meant to motivate researchers and practioners.

Categories and Subject Descriptors

H5.2 [**User Interfaces**]: *prototyping, evaluation/methodology, theory and methods.*

General Terms

Design, Experimentation, Human Factors

Keywords

Prototype, Development Lifecycle, Air Traffic Control, Model, Change

1. INTRODUCTION

Life cycles of user centred user interface development are well known and consist of eliciting user needs and their environment, specifying the user and organizational requirements, producing design solutions followed by evaluation, usually in several iterative cycles in an interdisciplinary team [8]. The four basic activities have been researched and practiced by developers often with good results.

How information flows between these four activities is not as well known and we hypothesize that this is the reason for the lack of interplay between evaluation and design. As in any activity, the four activities have input and output. The input is the basis for the activity and the output is the deliverable of the activity and usually input into the next activity in the lifecycle. The output of elicitation can be user, task or goal models (work models) of various types, and description of actors and their environment, i.e. context. The output of the design activity is one or several design ideas for a feature realized in low to high fidelity prototypes, a model or a final system. The output of the evaluation activity can be failures detected, hindrances, facilitators, and positive or negative consequences of a designed feature. The lack of means of translating output, coming either from elicitation or evaluation, to design ideas is an obstacle in the lifecycle of development of user interface.

If a design for a feature is rejected, it can be difficult to decide how it should be changed. Then we need to go back to the drawing board to create new design ideas. In software development, finding root causes has been widely used and the CUP (Classification of Usability Problems) [7] method has been suggested to further classify attributes of failures in user interaction and to find their roots in processes of the user interface development lifecycle. To find causes of problems (e.g. undesirable effects), i.e. backwards at the time of evaluation, we may record the cause and the desired effect at the time of design. The causes may be miscellaneous and even multiple; they can be within the design features or the underlying work model. Hence, one should also note the implications a work model is meant to have on design. (see Figure 1). In this paper, we set forth research questions that have emerged from our work in prototyping and evaluation of two case studies in air traffic control. The aim of presenting the case studies is to examine the activities and learn how they can be a basis for discussion of a development lifecycle and in particular its work products. The next section gives an overview of two design experiments where low-fidelity prototypes have been used. Examples in the remainder of the paper are taken from the case studies.

NordiCHI'04, October 24th, 2004, Tampere, Finland.

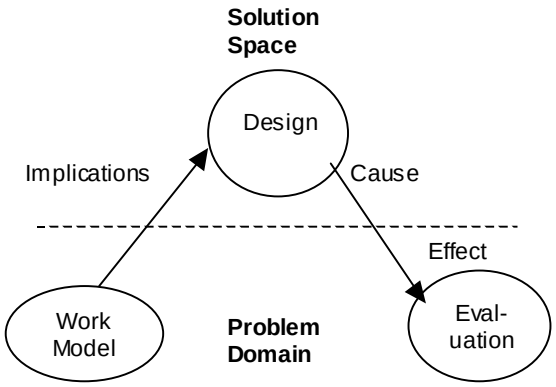

Figure 1 Cause in Design and Effect in Evaluation

Table 1 Methods used

	Speech Agent	**Integrated workstation**
Elicitation	Literature review Observation Interview	Observation Interview Existing systems & requirements studies Class & Collaboration diagrams Cognitive models of user's work Heuristics evaluation using cognitive principles
Design	Architecture Sequence diagrams Prototype	Paper sketches Three alternative approaches suggested
Evaluation	Wizard of Oz with air traffic controllers Post-test questionnaire Qualitative and Quantitative data gathered	Claims analysis Walk-through of drawings of user interface with participation of air traffic controllers post-task questionnaire Qualitative data gathered

2. CASE STUDIES

The two case studies reported here are taken from the domain of air traffic control. The duty of the air traffic controllers in the studies is to service aircraft en route in oceanic environments, i.e. cross the North-Atlantic. They monitor aircraft against predetermined routes, but issue clearances for requests for different routes provided it is safe, i.e. if aircraft adhere to separation rules. In the following two subsections, we describe how elicitation, design and evaluation were carried out in the two projects. Table 1 provides an overview of the methods used.

2.1 Using language technology to improve communication in ATC

2.1.1 Elicitation

Previous literature on voice communication in Air Traffic Control was analysed [10]. Oceanic Air Traffic Controllers were observed while at work at a centre of air traffic control and operators at a Centre of Radio Communication were observed. The researcher interviewed expert controllers to learn about the domain of Air Traffic Control and to understand the role of voice communication. The challenge in this domain is that fortunately errors in voice communication are relatively infrequent so they are not easily observed.

2.1.2 Design

A prototype of a speech agent was developed with the goal of recognizing errors in the communication between pilot and controller. Several options to replace or add a speech agent to existing voice communications were explored and their architecture designed, but a prototype of only one was implemented.

Sequence diagrams describing realistic scenarios, edited by expert users, of dialogues were created for three characteristic scenarios of the problem domain.

2.1.3 Evaluation

A Wizard of Oz evaluation was conducted with five controllers of varying expertise. The evaluation was scripted, using the dialogues described in the previous section, with the tester playing the role of the pilot against each of the controllers. Quantitative data was gathered on errors made by the speech server during evaluation and quantitative and qualitative data on controllers' attitude towards trust and performance was gathered in a post-test questionnaire. The evaluator asked questions about the type of feedback a speech agent should give in case of error in the voice communication between controller and pilot.

Since the prototype was of low fidelity, it was not feasible to evaluate it in context, other than to create real life scenarios and to have actual users. Evaluators were not conducted in Air Traffic Controllers' room or in a group with collaborators of the work, such as controllers of the same centre, supervisors, or controllers of adjacent centres. Although this is considered important, and perhaps especially so when researching voice communication, it would have been impossible to get permission for evaluation on site and hence would have to be staged.

2.1.4 Results

Since the tests had to be scheduled in advanced, and resources were scarce, there was no time to pilot test the evaluation on site. Hence, some of the evaluation instances were flawed because of failures in the supporting technology. The script worked very well, the performance of the speech agent was measured and the controllers were able to understand and reflect on the concepts. Controllers' attitude towards expected efficiency, safety and their trust on the speech agent has to be viewed in context of the artefact evaluated, but were acceptable to proceed to the next phase. Performance of the speech agent gave the designer good ideas on how to improve its design and implementation.

2.2 Integrating different user interfaces in a controller's workstation

2.2.1 Elicitation

As in the previous case study, observations were made, but a wider range of controllers was interviewed [9]. The architecture of different subsystems of a workstation was analysed including their relationships.

An abstract model of the problem domain was created based on manuals of operations, previous requirements studies, observation of work and current systems. The model was expressed with text and UML diagrams.

A user interface model was reengineered from two current systems in order to find possible anomalies and basis for integration of two user interfaces.

Cognitive models of user's work were examined.

Heuristic evaluation, using cognitive principles, was carried out on current ATC's workstation to find deficiencies.

2.2.2 Design

Three alternative approaches to integration were described but one of them designed in detail as drawings of user interfaces.

Snapshots of user interfaces of design ideas for several features were created in a drawing tool. Snapshots were ordered into a short storyboard explaining a scenario of work.

Except for the description of the integration of the three alternatives, no models of designs were made, neither as scenarios, interactions, navigations, dialogues nor structure of user interfaces. The reason may was that the focus was on limited design features illustrated at the presentation level.

2.2.3 Evaluation

Evaluation did not take place in context, except that interviewees were air traffic controllers. Controllers were asked to give a preference to one of three alternative approaches to integration of the two user interfaces. A researcher conducted claims analysis [11] of three alternative approaches to integration.

Evaluations of snapshots were made with controllers of varying expertise. No interaction took place but instead the researcher described situations to users. For some features several alternatives were presented and users asked to rate them and discuss, but for others only one design was presented. The method of evaluation was an interview with predetermined questions about safety, performance, and invited design suggestions from the controllers. Two iterations of evaluations took place with feedback from the former affecting the latter.

2.2.4 Results

The snapshots of designs of user interfaces provided valuable means for interviewing users about the new ideas. Researchers received good ideas from users and the two iterations showed that improvements were achieved. The triangulation of evaluation methods, i.e. claims analysis and users' preference gave researchers additional confidence in the results.

The abstract models drawn and the cognitive models examined during elicitation were both useful to understand the complex problem domain and to explore new design ideas for specific aspects. They were particularly helpful in moving away from current context, which was necessary because the technological and consequently other contextual layers are changing.

3. ELICITING NEEDS AND CONTEXT

In this and the following two sections, we describe the activities of the user interface development lifecycle. We end each section with questions or challenges that will help us link the activities. Prior to the questions, we give examples from the two case studies. The first activity in a human-centred design is to understand and specify the context of use. Contextual inquiries [3] and ethnographic approaches have been gaining popularity in recent years. Less is known about how to produce work products that are useful for software engineers or user interface designers. Context, partnership, interpretation, and focus are four principles that guide contextual inquiry. The first and most basic requirement of Contextual Inquiry is to go to the customer's workplace and observe the work. The second is that the analysts and the customer together in a partnership understand this work. The third is to interpret work by deriving facts, make hypothesis that can have implication for design. The fourth principle is that the interviewer defines a point of view while studying work. The output of this activity can be e.g. a work model and Beyer and Holtzblatt [3] suggest several models that comprise the work model, i.e. a model of communication, a sequence model, an artefact, or cultural and physical models. The lack of formalism in these models makes them difficult for practioners like engineers to adopt. Semi-formal models in UML could replace or complement these informal models.

Vicente [12] argues that work analysis for systems should identify and model intrinsic work constraints, and that the models should have formative implications for design. The motivation is that there is no systematic way to go from results of testing to prototype attributes and therefore we are dependent on the creativity of the designer to revise the prototype to remove the problematic effect. The CWA (Cognitive Work Analysis) is an example of such a formative approach to work analysis and so is the Contextual Design proposed by Beyer and Holtzblatt [12]. Above we listed the models of Contextual designs that are created, but CWA presents other conceptual distinctions [12, p. 120]: *Work Domain*, *Control Tasks*, *Strategies*, *Social-Organizational* and *Worker Competencies*. Through analysis of these distinctions, models of intrinsic work constraints are created that again lead to system design interventions. We give examples of interventions for Strategies, Social-Organization and Worker Competencies. Dialogue

modes and process flow are based on constraints derived from strategies. Role allocation and organizational structure are based on Social-Organizational constraints. Training and interface form are based on constraints derived from worker competencies. Neither Vicente nor Beyer and Holtzblatt express explicitly or maintain in a formal way the design implications of the work analysis. Vicente gives informal relationships in between the two activities by taking examples, but work analysis and not design is the subject of [12]. More often than not, motivations for system implementation are changes. Those changes are e.g. due to changing technological contexts of the problem domain, increased scale, increased demand for quality or changing technological changes in the solution space. Below, is an example that shows how proposed changes in Social-Organizational conceptual distinction has an implication to a design.

A simplified example from the speech agent
Social-Organizational: A speech agent replaces a radio operator.

How can the implications of work analysis to design be modelled and maintained?

4. DESIGN

The data collected during elicitation and evaluation of previous versions of the modified problem context will guide new design ideas. Design can be abstract such as re-design of work or structure of information, to detailed interactions between a product and the context. The design of the user interface is of this last type.

Before a user interface is programmed, we can create a model of the design that we use to evaluate against our requirement and assumptions. The model may range from being abstract, like diagrams or wire frame, or detailed, such as sketches. Prototypes of various types, i.e. low vs. high fidelity, experience prototypes [5], vertical and horizontal, throwaway and incremental prototypes, are popular since they give the user an idea about the look and the feel of the interface. Other products can be used to model certain aspects of a user interface such as navigation, dialogue or architecture such as diagrammatic models e.g. in UML or extensions thereof. Storyboards and textual scenarios are often useful to present design ideas or concepts respectively early on.

Designers should select the type of model that is most appropriate for the design feature at hand. For example, when designing complex navigations, a navigational diagram that gives an overview of the traversals between contexts will be more useful than many detailed sketches of designs. On the other hand, when designing presentations for entities that contain a rich collection of information, sketches are more useful. A complex dialogue implementing a scenario may be best presented with both sketches and diagrammatic models.

Speech agent: The Wizard of Oz prototype was supported by a sequence diagram describing scenarios that were evaluated.

How can we guide designers to use a combination of different design products, such as different fidelities of prototypes, diagrammatic models, text scenarios, or text use cases?

4.1 Multiple Design Ideas

One of the fundamental principles of design is to create multiple design ideas for a feature. This can be a result of a brainstorming session with an interdisciplinary team including users. When the design team has been a participant in the whole lifecycle, design ideas are implicitly linked to user needs and context of work.

The rationale for the design idea needs to be made explicit. Otherwise it will be difficult during evaluation to validate weather the design feature is coherent with the problem domain. Evaluation of the design is prepared during the design phase. In our experience, it is not adequate to ask whether a design meets requirements of efficiency, effectiveness and satisfaction especially for design ideas produced early in the life cycle, often in low fidelity prototypes. Designers should associate evaluation questions with the design ideas during design but not after it. Thus, usability specialists should either work on the design team, or in small organizations designers, should take on the role of usability test designers.

Integration of ATC: Three alternative design ideas for integration were presented. Claims analysis was applied to elicit positive and negative consequences. Questionnaires were posted to elicit views on usability of the alternatives.

How can we design and describe evaluations of user interfaces that can answer specific questions about the effect of the design?

4.2 Tradeoffs

Design ideas are created to change a problem domain. There may be different motivation for the change, i.e. technical, social, organizational, or economical. Common effect of the changes that we are aiming for are increased effectiveness, efficiency or satisfaction during operation. Other changes may result in increased safety or less time for training. A design idea that may cause a positive effect of one aspect of the problem domain may at the same time cause a negative effect of another. We take an example from combining two user interfaces, Flight Data Processing (Flight strips) and Radar Data into one. The merging of the two interfaces will eliminate the need to integrate information in the user's head but can increase clutter on the display. More automation in the Flight Controller workstation can lead to less workload in easy low traffic situations but may blur the controller's mental picture (leading to less efficient or safe operations) during difficult high traffic or critical situations. The above statements are similar to claims analysis [11] where positive and negative consequences of a single design feature are gathered. Bass and John [1] describe how we can analyse tradeoffs of different software architecture patterns and their effect on usability.

Integration of ATC: Controls for selecting altitude levels cause the controller to focus on specific critical air traffic and reduces the cognitive load, thereby making decisions easier.

How can we express the expected effect in the problem domain, resulting from changes brought on by the design ideas?

Integration of ATC: Controls for selecting altitude levels cause the controller to miss information in deselected altitudes and therefore deteriorating the mental model of the current state of the system.

As we see above, when creating different design ideas, there can be tradeoffs between them. Another example is taken again from ATC. Either adaptable (i.e. adapted by the user) or adaptive (adapted by the computer) user interfaces are meant to solve the problem of display clutter that can occur during high traffic situations.

Integration of ATC: An adaptive interface can be more efficient than adaptable interface to the controller but less satisfying.

How can we express tradeoffs of effects between design ideas?

5. EVALUATION IN CONTEXT

The goal of the evaluation is to see how the proposed changes interact with the problem domain. Hence, by introducing the new design, we have modified the problem domain.

Many methods of evaluation have been proposed, both analytical and empirical, most are manual but some are also automatic. The results are either qualitative but also quantitative. Evaluations are done at different phases in the development life cycle, but close interaction with users from an early stage has been advocated. Evaluating finished products may be easier but failures detected at such a late stage may be costly to correct. Hence, designers have focused on early evaluation with low-fidelity prototypes, experience prototypes with users. The down side is that these evaluations may not be as reliable e.g. in safety critical situations.

Although contextual inquiries have been advocated, there has been less emphasis on evaluations of design in real contexts and, in the case of early evaluation, this may prove to be unfeasible. However, experience prototyping [5] has been proposed as a tool to use for this purpose. Every effort should be made to place the design in real contexts. Training facilities can be used to accomplish this. Simulators may be another way.

Speech agent: Controllers were recruited to participate in the evaluation, scenarios were carefully designed and verified to emulate real contexts.

How can we build a context for evaluation of designs during early phases?

The results of an evaluation can be twofold; either the design ideas were not able to correctly fulfil the assumptions or the model of needs or the underlying model of the problem domain proved incorrect. This results in changing the model or changing the design. In the former case we might have assumed something about the work or its context, but found out during subsequent evaluation that the assumption was not correct. An example from the speech agent is that we assumed that controllers spoke at a specific speed with no delays. This assumptions lead to a certain configuration in the agent. This is an example of a relationship between how some knowledge about he problem domain leads to a design decision. During evaluation, it became evident that the assumption was not correct. If we have a model of the relationship between the problem domain and the new design ideas, it will be easier to trace back the causes of failures, correct the underlying model, and adapt the design. Not all such relationships may be evident beforehand and some are only realized during evaluation.

Although we may have specified the expected effect of a design idea, it may be that it will lead to some actual unforeseen effect. The evaluation in context is about finding out how the new design ideas interact in the changed problem domain. Hence we try to observe what changes the ideas bring about to the entire problem domain, not only the immediate user, but other systems and stakeholders.

How can we express the actual effect in the problem domain, resulting from changes brought on by the design ideas?

If we fail to reach the desired effect, we may either trace backward to the documented causes of the desired effect or else we need to trace it to failed designs or wrong assumptions in the problem domain.

How can failures (in reaching the desired effect) lead us to failed designs (causes) or wrong assumptions in the work model?

6. DISCUSSION

This paper has presented challenges that need to be addressed to better integration evaluation and design. The approach proposed involves specifying different work products. We have used two case studies to illustrate our challenges with simple examples, expressed above in boxes. They are by no means meant to be examples of how to address these challenges, but rather give some initial illustration of the concepts.

Although it is easier for developers to understand the lifecycle consisting of separate activities and we understand that it is important to have several iterations of the activities, the gap between them may be unnecessary.

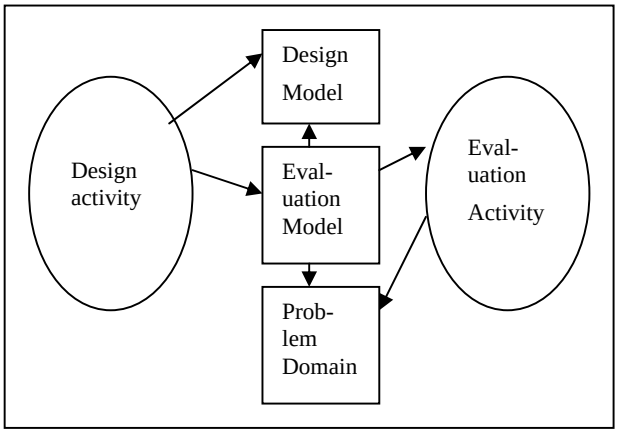

Figure 2 Development model

We propose (see Figure 2) to have two activities and that the Design and Evaluation activities are run concurrently, with the two artefacts Design (and/or a model thereof), the Model of the Problem Domain, and The Evaluation Model as central repositories. The distinction between elicitation and evaluation may not always be clear since evaluation elicits new information and gives us further data about user needs and their

environment. The only difference between them is that at elicitation usually (but not always) no design of features is presented. This constitutes the first iteration, but in subsequent iterations, we use the term evaluation because some product of the design has entered the domain.

The Evaluation activity should not be conducted as a separate activity after the Design, but instead planned for during Design and then carried out. We have a practice in software development where it is recommended to design the test before the implementation. Extreme Programming [2], which is a type of an agile development methodology, has this practice as one of its main guidelines. Cockburn [6] offers two advantages of automated regression tests: the developers can change the code and retest it very easily and there is less stress if the developers can run automated regression tests since they are then ensured that no one else has altered the code. Unfortunately, it is difficult to write such automated test for user interfaces, and hence the more reason to attempt to make them formal and easily repeatable. Briand et al. [4] have proposed a revision of the Goal Quality Metric framework, called GQM/MEDEA that adds empirical hypotheses and aims to make them quantitatively verifiable.

7. ACKNOWLEDGMENTS

Margrét Dóra Ragnarsdóttir and Hlynur Jóhannsson have designed and evaluated the prototypes of the speech agent and integration of user interfaces respectively.

8. REFERENCES

[1] Bass, L., John, B. Linking Usability to Software Architecture Patterns through General Scenarios, *The Journal of Systems and Software, 66 (2003) 187-197*

[2] Beck, K., Test-driven development, Addison-Wesley, 2002

[3] Beyer, Hugh and Holtzblatt, Karen, *Contextual Design*, Morgan Kaufman, 1998

[4] Briand, L. C., Morasca, S., Basili, V. An Operational Process for Goal-Driven Definition of Measures, *IEEE Transactions on Software Engineering, vol. 28, no. 12, December 2002*

[5] Buchenau, M., Suri J. F., Experience prototyping, DIS'00, ACM, 2000

[6] Cockburn, A., *Agile Software Development,* Addison-Wesley, 2002

[7] Hvannberg, E.T, Law, L. C., Classification of Usability Problems (CUP) Scheme, Interact'03, IFIP, Switzerland, 2003

[8] ISO/IEC 13407:1999 Human-centred design processes for interactive systems

[9] Johannsson, H., Hvannberg, E.T., Integration of Air Traffic Control User Interfaces, 23rd DASC, Digital Avionics Systems Conference, IEEE, 2004

[10] Ragnarsdottir, M. D., Waage, H., Hvannberg, E.T., Language Technology in Air Traffic Control, 22nd DASC, Digital Avionics Systems Conference, IEEE, 2004

[11] Rosson, M.B. and Carroll, J. Usability Engineering: Scenario-Based Development of Human Computer Interaction, Morgan Kaufmann, 2002

[12] Vicente, Kim J. *Cognitive Work Analysis*, Lawrence Erlbaum associates, 1999

Value Enabling Interaction Mediates Between Design and Evaluation

Gilbert Cockton
School of Computing & Technology,
Sir Tom Cowie Campus, University of Sunderland,
St. Peter's Way, Sunderland SR6 0DD, UK.
Gilbert.Cockton@sunderland.ac.uk

ABSTRACT
The interplay between usability evaluation and user interface design is indirect and must be mediated by value enabling interaction. We do not evaluate systems in HCI, we evaluate interaction. We thus cannot evaluate designs, but only their consequences for the quality of interactions. In evaluating interaction, we anticipate or observe user difficulties. A design may or may not contain the potential causes of a user difficulty. Causes have to be inferred from user difficulties in context. There is thus no direct interplay in either direction, either from design to evaluation, or from evaluation to design. Instead, both are mediated by interaction, but even this mediation is not direct. We must reason from designs to interactions, and from interactions to design features as causal factors. However, these processes are inherently descriptive. The role of evaluation must go beyond description to judgement, since the literal meaning of "evaluation" is to *(bring) out value*, that is, to find it in one place and to express it somewhere else. In HCI, we find value in interaction, but we judge value in the world. Until we start by stating the intended value of digital products, HCI can not reach the end point of delivering computer systems that are *worth* using. The relationship between design and evaluation is thus mediated by user interactions that do (not) deliver intended value.

Categories and Subject Descriptors
ACM: H.1.2 – User/Machine Systems

General Terms
Design, Economics, Experimentation, Human Factors,

Keywords
Value-centred HCI, Design, Evaluation, Mediation.

1 INTRODUCTION
The workshop title "Improving the Interplay between Usability Evaluation and User Interface Design" implicitly, if not explicitly, assumes a direct relationship ("interplay") between design and evaluation. No such direct relationship exists. In reality, the relationship must be, not *interplay*, but a *chain of mediations* via user interaction and the intended value for a digital product.

NordiCHI 2004 Workshop on Improving the Interplay between Usability Evaluation and User Interface Design, October 24, 2004, Tampere, Finland.

2 VALUE AND E*VALU*ATION
The English word *evaluate* is a back-formation from the French *évaluation*, which in turn is formed from the French *évaluer*, that is *é+valuer*, which literally means to (bring) value out of (from the Latin prefix, *ex*, which here became *é*).

Value and e*valu*ation are this inextricably linked, and it is thus somewhat unnerving that this has hardly been mentioned in over three decades of HCI research and practice. The sense of "evaluation", like many English words, has broadened, so that the Freesearch on-line dictionary [9] defines it as:

> ***to judge or calculate the quality, importance, amount or value of something:***

The tendency within HCI has been to see evaluation as mostly a question of quality, sometimes of degree (amount) and of importance, but rarely of value. However, I will argue that it is possible to have quality with neither value nor importance, especially where quality is assessed with respect to generic standards and measures (amounts of errors etc.).

Interestingly, the Concise English Dictionary [12] has a weaker first sense for *evaluation* as "assess, appraise" and a second mathematical sense as calculation of some form. For many who use the word, "evaluation" has lost its clear connection with "valuation". I will argue that effective evaluation in HCI should be understood in terms of intended value for digital products and services. Value here is not necessarily commercial. It can be personal, spiritual, experiential, organizational, political or cultural. Value-centred HCI must thus be able to cope with a wide range of human values. The core skills here are the ability to express intended value, the ability to relate this via envisaged interaction to design decisions, and the ability to relate value to the planning and interpretation of system evaluation.

3 THE ARGUMENT FOR VALUE-CENTRED HCI
The argument is a historical one. We have exhausted objective and descriptive approaches to HCI. Over three and a half decades, we have moved through three foci for HCI: the system, the user and the context of use [4]. None of these can function adequately as the sole focus for HCI [3].

3.1 The System as Focus

Early HCI work focused on design guidelines. This tradition has continued, and many still act as if universal "one size fits all" solutions are possible for interactive systems. Such design rules are rules about system features. The assumption is that such features can be directly evaluated, but, from an HCI perspective, they cannot be. In HCI we evaluate interaction. It is difficult to imagine what evaluating a system could mean in human terms. The attributes of systems that can be directly evaluated concern internal, rather than external quality, that is, qualities such as performance efficiency, correctness, modifiability and maintainability [10].

A system or design can be described. Claims (often wild and unrealistic) can be made for systems or designs. However, in HCI, we can only evaluate *usage*. We look at the interactions between people and systems. While this may be obvious to evaluation experts, it does not stop people outside of HCI (and too many within) from acting as if designs can be evaluated and that quality can be encapsulated in good features.

We need to understand how such an illogical situation persists, i.e., a belief in quality *within* a digital artifact rather than the user experience. The origins of the belief may lie in the origins of computer science. These are more than harmless philosophical concerns: they lead to damaging technological utopianism and a fetishism of technology alienated from its human context.

A system-centred approach is a natural consequence of the mathematical Platonic mind-set in Computer Science. Many mathematicians believe in mathematical *discoveries* on the basis that there is a single fixed mathematical reality that is revealed through mathematical investigation. Mathematical objects, although wholly abstract and apparently constructions of human imagination, are held to *exist*, almost in the sense that physical objects exist, except that they cannot be directly perceived (i.e., they are not sensuous). These ideal forms have fixed inherent properties that are the essence of mathematical objects.

This Platonic view has severe consequences for HCI when transferred to computer systems, since mathematically inclined technologists are inclined to treat software as a mathematical object with fixed inherent properties. This manifests itself in HCI in the form of design principles, patterns and guidelines. While these can be contextualized, the overwhelming tendency is for design principles, patterns and guidelines to be stated as "one-size-fits-all" absolutes. The result is that human agency, individual differences and usage contexts are removed from the equation. This isolation, or estrangement, of humans from the properties or *qualities* of computer systems is a form of alienation, which has some of the key consequences outlined by Marx in the Paris Manuscripts [11]. Systems are described as *fetishes* with *totemic* qualities, just as commodities become fetishes by the alienation of human labour from its products. Marx's analysis is quoted and summarized as follows [8]:

> "A commodity appears at first sight an extremely obvious, trivial thing. But its analysis brings out that it is a very strange thing […]." Fetishism in anthropology refers to the primitive belief that godly powers can inhere in inanimate things (e.g., in totems). Marx borrows this …to make sense of what he terms "commodity fetishism" … the commodity remains simple as long as it is tied to its use-value. When a piece of wood is turned into a table through human labor, its use-value is clear and, as product, the table remains tied to its material use. However, as soon as the table "emerges as a commodity, it changes into a thing which "transcends sensuousness"… People … thus begin to treat commodities as if value inhered in the objects themselves, rather than in the amount of real labor expended to produce the object. What is … a social relation between people … instead assumes "the fantastic form of a relation between things".

We see very similar processes in operation with system-centred HCI. Once quality is seen to reside in systems, magical claims follow thick and fast. Within the history of HCI, we have been told that graphical user interfaces were inherently easy to use, that on-line agents will solve all our shopping dilemmas, that location-based services will bring us desparately sought information. In all cases, the new technologies will automatically deliver a technical utopia in all contexts for all users. The consequences of computer science thinking are explored further in my NordiCHI plenary [5].

To some extent, the first two questions for the workshop construct design products as things with intrinsic properties:

(1) Which products of interface design are useful as the basis for usability evaluations?

(2) How do the specific products from interface design influence the techniques that are relevant for the usability evaluation?

The answer to the first, given that we cannot directly evaluate systems, is "none". The answer to the second is that "they should not". We evaluate interaction, and what we thus require from design is the ability to contribute to the direct evaluation of interaction. There are two forms of design products that can do this. Firstly, some can be tested with users, such as paper mock-ups, wire frames or prototypes of varying fidelity. Secondly, some can be combined with contextual research and HCI knowledge to produce models or descriptions of potential interaction, which can then be evaluated (e.g., task models for GOMS or task descriptions for Cognitive Walkthrough).

What is key about design products is how well they let us create actual or imagined interactions. Actual interactions arise when evaluation participants interact with design products. Imagined interactions arise when we derive interaction sequences from design products. As long as we can create actual or imagined interactions, then design products are compatible with evaluation. The quality of evaluation depends in part on the quality of the created interactions, but the key to evaluation is understanding value, and this is wholly independent of design products. Value pre-exists and post-endures design and interaction. It should thus be possible to plan much of evaluation before any design product at all exists in any form.

In summary, system-centred HCI is illogical. Systems cannot be evaluated, only interaction can be. To support evaluation, design products must be able to either produce real interaction, or support the synthesis of predicted interactions. There is no direct interplay between design and evaluation. Both must be mediated by actual or imagined interaction. What we thus require are methods that situate design products within usage

interactions. Not surprisingly, this is how HCI evolved in the 1980s.

3.2 The User as Focus

System-centred HCI was succeeded by user-centred HCI. User testing and inspection methods were a key part of this progression. Usability evaluation came to focus on quality in use. Users' difficulties when interacting with a system would be observed and described. User-centred HCI moved from misguided attempts to evaluate software systems to evaluating the quality of interaction associated with a specific design. However, user-centred HCI doesn't really *evaluate* interaction, nor can it always link back its 'results' to design features.

Usability engineering approaches rarely *really evaluate* since they have no concept of intended product *value*. They thus cannot properly prioritize user difficulties. Generic severity scales (e.g., [14]) are not appropriate. Thus one may think that task failure is always the most severe form of "usability problem", but severity here actually depends on how critical the task is to delivering the intended value of a digital product. In some contexts, task success with residual errors (e.g., in the design of a safety critical product) is more severe than task failure. Non-existent designs are infinitely less harmful than dangerous ones.

Usability engineering tends to be context-independent. While user test scenarios may attempt to recreate real contexts of use, the results of user testing may not be reported back in a contextually sensitive manner. Error counts, time on task, success rates and subjective response can be treated as universally relevant measures. So in answer to the question:

(3) In which forms are the results of usability evaluations supplied back into interface design?

The answer is often "a useless one", i.e., the results of user testing take no account of what a product or service is trying to achieve. While in practice, experienced usability specialists do take business and other client goals into account when reporting, no existing method makes clear use of statements of intended value as inputs to planning and reporting. SUPEX [6] provides a 'filtering' hook to accommodate product goals, but no more. The result is that our publicly documented methods only aim to deliver quality in use. Achievement of product value is a matter of luck.

Without explicit inputs for intended value, usability engineering methods cannot be seen as *evaluation* methods. They assess and appraise in the weak sense of "evaluation", but they are not focused on establishing the impact of the user experience on the achieved value for a digital product or service. This impact occupies a continuum from destruction to donation. User experience may be so poor as to *destroy* all intended value. Conversely, it may be so surprisingly good that it *donates* unexpected value, i.e., both product sponsors and customers get more than they expect, which is the true mark of *gifted design*.

In between destruction and donation, user experience may *degrade* or *deliver* intended value. Where we cannot understand or fix catastrophic problems (because the technology simply can not work as hoped), then we must *deny* the possibility of a design ever delivering its intended value. Such reality checking is a common role for human factors experts, especially in response to naïve technological utopianism. However, denial is based on the absence of credible fixes and thus goes beyond the scope of evaluation. It is rather an issue for the *iteration* of designs.

There are thus '5 Ds' of HCI: deny, destroy, degrade, deliver and donate. The last four are a basis for *value impact analysis*. They do not feed through directly into design, and nor does the first, other than stopping all further design.

The purpose of evaluation is to assess value. It is not the role of evaluation to propose design changes. We need to be clear in distinguishing the description of interaction (the 'results' of user testing) from its evaluation (which assesses impact of user interaction on achieved product value). Evaluation results are thus not directly 'supplied back' into design. Instead, they isolate and identify the user difficulties (actual or imagined) that really matter. Design change recommendations need to be based on a credible causal analysis of user difficulties. This is *not* part of evaluation. The purpose of evaluation, once again, is to assess achieved value. *Explaining* why value is or is not achieved is a very different activity to *assessing* the achievement of value.

Thus in response to:

(4) Which usability evaluation results are needed in interface design?

The initial answer is that *value impact analysis* will identify user difficulties that destroy or degrade the achieved value of a digital product or service. However, progressing from this identification to the design changes that may deliver or donate value requires two distinct steps that are not part of evaluation activities. Instead, they are part of the *iteration* activities that move a design from one combination of value to a (hopefully) improved one. Dennis Wixon limits effective evaluation to two questions [1]: Do we understand the problem? Can we fix it? Reports of user difficulties are not sufficient for either. However, neither of these questions are part of the evaluation process, which should stop with identification of user difficulties that degrade product value.

User-centred HCI has provided little systematic support for iteration, which requires two distinct activities. The first is *causal exploration and analysis*, which may require more formal studies (even controlled experiments) to establish the causes of value degrading user difficulties. Evaluators need to work in collaboration with developers to properly structure causal analysis (often a developer will immediately understand why a difficulty has arisen, but an evaluator could take hours to reconstruct a causal chain).

The second iteration activity is *design change recommendation*, which requires extensive knowledge of interaction design and a full understanding of the goals for a product or service. An evaluator may not have all the knowledge and skills required to make credible design changes without the collaboration of software and project specialists.

We should thus separate evaluation from iteration. Evaluation should report in terms of value, and not in terms of generic error counts, stories of unhappy users and time on task — except

where these measures and information have a direct bearing on intended value.

Design iteration requires not only confidence in the results of usability evaluation, but also information that is directly relevant to making design decisions. Observations of user difficulties are only part of the analysis. Re-design requires a sound understanding of how design features combine with usage contexts to degrade the user experience.

In summary, current usability reports are not well focused on value, nor do (or should) evaluation methods be the main palce for causal analysis that can directly identify how users and design features interact to produce (un)acceptable interaction. Evaluation methods fail to provide what is needed, which is an evaluation in terms of 4 Ds of HCI (destruction, degradation, delivery and donation). Evaluation ends here. Iteration begins with the search for explanations of the impact of interaction on product value in terms of causal chains between user behaviour and system features. Iteration may end with the denial that intended value can be achieved with a target technology.

Dissatisfaction with 1980s user testing approaches [13], especially overreliance on generic measures such as error counts and time on task, led to the next major paradigm shift within HCI. However, while the move from system to user led to significant progress within HCI, the move from user to context did not address the main requirements for true evaluation: a focus on value.

3.3 Context of Use as Focus

The move from user- to context-centred HCI in the 1990s enabled more contextually sensitive and appropriate evaluation measures. These were (and are) more suitable inputs to value impact analysis.

The focus moved from the minutiae of quality in use to major issues of the fit between a design and an intended context of use. Contextually realistic evaluation increases confidence in the validity of reported user difficulties, but it does not move testing from appraisal/assessment to true evaluation.

Contextualised descriptions of user difficulties and interaction misfits are broader and more specific, and thus provide a conceptually richer space for explanation. As a result, context-centred HCI is better placed to understand what it is that intended users will actually value. Contextual research can be focused on understanding value in a way that psychological laboratory testing cannot. However, context-centred HCI has tended to focus on the 'fit' between a design and its intended context of use [3]. As with user difficulties, not all misfits have major consequences for the delivery of intended product value.

Context-centred HCI, as with user-centred HCI, lacks critical 'noise filters' that will focus evaluation on the delivery and enhancement of intended value. In electronics, a noise filter (such as Dolby™ tape noise reduction) removes noise from a channel, leaving mostly signal. We need something similar in HCI to isolate important problems from 'noisy' usability 'non-problems' and trivial inconsequential misfits.

In summary, contextual approaches have focused on (mis)fit, without necessarily addressing value or importance. Even so, they do identify potential loss of value that cannot be identified by traditional user testing. We must thus see evaluation as not only focussing on quality in use, but also on fit to context.

3.4 Answering Other Workshop Questions

From the above, my answers to the remaining questions could be very predictable:

(5) Do existing evaluation methods deliver the results that are needed in user interface design?

(6) How can usability evaluation be integrated more directly in user interface design?

(7) How can usability evaluation methods be applied in emerging techniques for user interface design?

My answers are:

(5) No, and it's not their role to. This is the role of iteration processes within development.

(6) Design and evaluation need to be integrated within a wider value-centred framework for HCI. Iteration is one key link between evaluation and design. The initial link is provided by *opportunity identification* processes. However, designs may need to incorporate support for evaluation.

(7) They cannot, evaluation and design need to be integrated within a wider value-centred framework for HCI.

This paper thus reframes the workshop problem. The relevant questions are not about direction relationships between design and evaluation, but instead about how design and evaluation relate to iteration and initial development in a value-centred framework,

4 A FRAMEWORK FOR VALUE-CENTRED HCI

The workshop focussed on the interplay between design and evaluation. In analysing the relationship between them, we need to bear the following in mind:

(1) Design and evaluation are complex processes that each require co-ordination of discrete activities

(2) We evaluate interaction, not design, so one of the interfaces between design and evaluation activities is the generation of actual or imagined interactions

(3) Evaluation should focus on the achievement of intended value, and thus statements of intended value are another interface between design and evaluation activities

(4) Evaluation does not (and should not) generate design recommendations, which are again the result of a process of co-ordinated *iteration* activities that begin with identification of destroyed and degraded value and end with design change recommendations

(5) Evaluation planning can commence once statements of intended value are available. Initial evaluation activities can be completed before any design activity commences. Only planning of precise evaluation procedures requires design products to establish the fine detail of evaluation.

(6) There are thus four broad processes in interactive systems development: design, evaluation, iteration, and *opportunity identification*, which must be completed before design and evaluation can begin. Similarly, iteration follows the completion of evaluation.

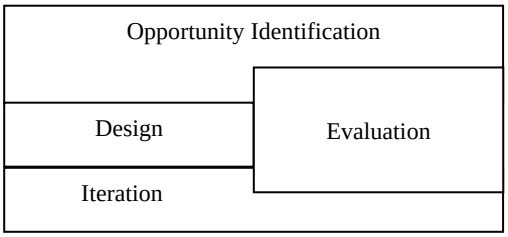

Figure 1. Main process structure for interactive systems development

Figure 1 shows the relationship between these development processes. Horizontal relationships indicate processes that can proceed in parallel. Vertical positions indicate logical dependencies, i.e., a process instance above must complete before the process below can start. Thus design and evaluation require statements of intended value as an output of opportunity identification to commence. Evaluation can start before design, and cannot complete before design (as this must pause to allow evaluation to take place). Similarly, evaluation must complete to allow iteration to commence. Once iteration is completed, development can recommence with new design and evaluation instances, or even with a revisitation of opportunity identification.

Figure 2 shows the internal structure of the four development processes within the context of a value-centred framework.Boxes represent products of development activities. Arrows represent activities which generate new development products from existing ones. Figure 2 is simplified. Arrows are labelled (e.g., E1), but not are all shown. For example, there should be arrows for activities that make use of *Design Change Recommendations* to update *Interaction Designs* and *Value Delivery Scenarios*, since causal analysis may have revealed poor decisions in any previous design activity. Similarly, causal analysis needs to be grounded in information on usage contexts, so a long back arrow is missing here.

Each arrow represents an activity performed by a development role. Thus statements of intended value are *derived* from representations of the context of use (activity O2), and are in turn inputs to both the *creation* of value delivery scenarios (activity D1) and a *transformation* into evaluation criteria (activity E1).

Activities in each main process are now briefly outlined. Examples are given for two hypothetical web-sites: one for van hire, and one for a university. Both are based on real development activities in which I have been engaged. The former involved commercial usability evaluation, and thus cannot be reported in any detail. The latter is ongoing and thus cannot be used as a detailed case study until further development iterations have completed.

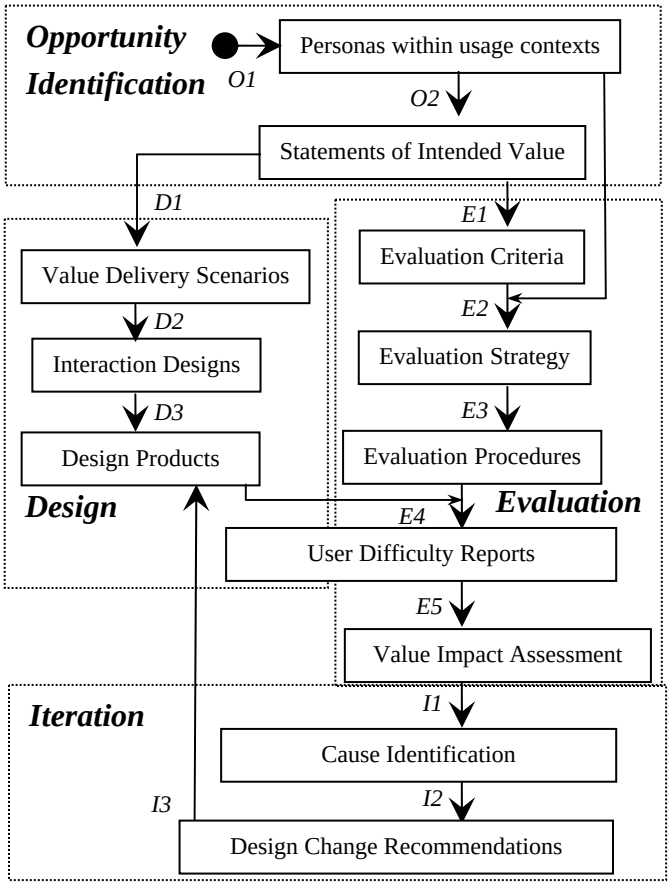

Figure 2. Activities and outputs within development processes for value-centred design

4.1 Opportunity Identification Activities

Opportunity identification is the process by which the intended value for a digital product or service is described and specified. It begins with *studies of usage contexts* (O1). These activities result in collections of models and descriptions of target usage contexts. The specific reference to personas [7] is deliberate. While most development products are general and should be able to accommodate a range of HCI methods, personas are highlighted as a method that are well suited to expressing the values of individuals and their organisations. Culture diagrams from Contextual Design [2] may also be appropriate forms for expressing value. Such development products are thus the main input to the second activity within opportunity generation: *intended value specification* (O2), which analyses contextual models and descriptions to identify opportunities for creating new value with a digital product and/or service(s). The result is a set of statements of intended value that should be delivered by a successful project.

For a university web-site, the key personas are university management, students, parents and career advisers. The primary value for the last three personas is the provision of appropriate, adequate and effective help with choice of course and university. For university management, a primary value from the web-site will be the achievement of high levels of student recruitment.

For a van hire web-site, the key personas are company managers, customers and depot staff, who respectively will derive value from: increased profits, improved brand equity, and recognised personal achievement; hiring an appropriate van for a suitable period at an economical cost as regards price and personal effort required to collect and return it; the smooth collection and return of vans by well-informed, well-prepared and satisfied customers.

The "intended value statements" for the two imaginary web-sites are very brief (e.g., "hiring an appropriate van for a suitable period at an economical cost as regards price and personal effort required to collect and return it"). The format and detail required for intended value statements is an open question in value-centred design. The sketches above are not sufficient, but at the same time, formats should be accessible for all stakeholders and the extent of detail is likely to be quite limited. Value can be stated succinctly.

4.2 Evaluation Activities

The evaluation process can begin before the design process, although both can run in parallel. The first activity, *value operationalisation* (E1) translates intended value statements into measurable evaluation criteria.

For a university web-site, example measurable criteria are the extent of engagement from site visitors (which pages get visited by who and what do they do as a result?) and the number of student enquiries, applications and enrolments that can be attributed to the university's web-site.

For a van hire web-site, example criteria are increased profits and improved brand equity attributable to the web-site. Although management desire recognised personal achievement and consequential career advancement, this is unlikely to be carried forward as an explicit evaluation criterion for a range of reasons that are easy to imagine. Further evaluation criteria are high levels of customer and depot staff satisfaction.

None of the example criteria above are usability requirements as would normally be understood. This is because quality in use and fit to context only matter in so far as they donate achieved value beyond what was sought, or when they destroy or degrade achievable value. Also, some of the example criteria cannot be addressed in existing usability testing approaches. Instead controls and measures must be placed *in the world, where value is achieved*, and not in the usability environment, which is transient and often artificial.

The second evaluation activity, *evaluation strategy formation* (E2), translates evaluation criteria into a strategy for monitoring and measuring the achievement of value. The main decisions here concern the choice of evaluation methods. User testing will be one part of this strategy, but evaluation has to extend to continuously monitoring the effectiveness of a system in real usage.

The third activity, *evaluation procedures design* (E3) selects measures and instruments for evaluation criteria that are appropriate for the evaluation methods selected as part of the evaluation strategy. Selected measures and instruments are associated with detailed procedures for each evaluation method.

The fourth activity, *evaluation implementation* (E4) applies evaluation methods to design products to produce reports of actual or predicted user difficulties. The fifth activity, *value impact analysis* (E5) assesses user difficulties in terms of their impact on achieved value. Only difficulties that destroy or degrade achieved value are carried forward for remediation during the iteration process. Note that value impact is not the same as severity. Most existing severity ratings are defined from a user/task perspective (e.g., [14]). However, value impact analysis has no pre-conceptions on whether task failure is always serious (it depends on the criticality of the task for delivering intended value), nor may moderate user disapproval be of limited concern (solid user approval may be vital to product success). What does and does not matter at this point is wholly dependent on earlier statements of intended value and their translation into evaluation criteria.

4.3 Design Activities

The first design activity, *value delivery scenario authoring* (D1) is similar to activity E1 (value operationalisation), as it restates statements of intended value in a form that can be used directly and effectively within subsequent activities in the design process. This activity refocuses existing HCI uses of scenarios to focus on the delivery of value in the world, rather than on quality in use and/or fit to context. It is guided by evaluation criteria that should be in place before scenario authoring is well advanced. Good scenarios here will be ones that tell plausible stories of how value results from envisaged designs.

For a university web-site, value delivery scenarios would explain how a proposed design would deliver appropriate, adequate and effective help with choice of course and university, and how this in turn would achieve high levels of student recruitment. Furthermore, once evaluation strategies are in place, value delivery scenarios should cover the details of how evaluation procedures will confirm the delivery or better of intended value. Thus the effectiveness of web content could be demonstrated via enquiry codes that link the web site into a university's marketing processes. Also, interactive content and downloads on the web-site could track prospective students from initial interest to making an on-line application. It would be possible to measure the attractiveness and effectiveness of web-site content. Usability evaluation would focus on quality in use, looking for interactions that degraded or destroyed intended value. The latter could indicate that the value delivery scenarios were misguided. Iteration would have to address this by changing scenarios as well as the design.

For a van hire web-site, value delivery scenarios would provide plausible stories on how proposed designs could increase profits and improve brand equity by letting customers hire an appropriate van for a suitable period at an economical cost as regards price and personal effort required to collect and return it. Other scenarios would tell stories of how site features ensure the smooth collection and return of vans by well-informed, well-prepared and satisfied customers.

With value delivery scenarios in place, the *interaction design* activity (D2) would create a set of interaction designs that would be used in the third *design implementation* activity (D3) to create design products. Design then halts until the evaluation and iteration processes have completed.

4.4 Iteration Activities

Iteration begins with *Causal Analysis* (I1), which seeks to identify the causes of user difficulties that destroy or degrade achieved value. The second activity *Design Change Recommendation* (I2) uses identified causes to generate design changes that should remove undesirable user difficulties.

Iteration activities require the involvement of all roles in development. Developers and designers need to support evaluators in the identification of causes of user difficulties. Evaluators' skills are of particular importance when further user testing or formal user studies are required to reliably identify the causes of user difficulties. The quality of identified causes is critical to recommending appropriate design changes. A change based on a faulty causal analysis is likely to not improve a design, and may even make it worse.

Designers, developers, marketing and product management need to be involved in design change recommendation. This is not a job that evaluators can carry out in isolation. Designers may have several untried options that could be tried for the next version of a design. Developers can identify the costs of various proposed changes. Marketing and product management can advise on the appropriateness of proposed changes in relation to the vision and goals for a product or service (i.e., they may be best placed to interpret intended value statements and relate these to proposed changes).

Change recommendations apply to all products of the design process. Scenarios, design rationales and details, as well as implementations, may need to be changed. The third iteration activity, *design change implementation* (I3), implements all necessary changes to any design product.

It may the case that no design recommendations can be made that can plausibly result in better delivery of intended value. In these situations, a project may have to be terminated. The possibility of achieving intended value is *denied.*

The outcomes of iteration are thus one of the following:

1. the addition of value to the outcomes of interacting with a digital product or service (an improvement on the donation of value, moving from delivery to donation of value, moving from destruction/degradation of value to degradation or delivery)
2. the termination of a project (the denial that intended value can be achieved through an apparently promising technology)

The 5 Ds of HCI can thus be used to assess not only the impact of user interaction on achieved value, but also the outcome of iteration and its associated design change recommendations.

5 CONCLUSIONS

There is no direct relationship between design and evaluation, which are complex, multi-activity processes that are mediated, initially by a process of opportunity identification, and lastly by iteration. Design and evaluation can proceed in parallel, but design will benefit from a timely consideration of evaluation criteria and evaluation strategies, especially when the latter embed evaluation instruments in the product.

The view that designs can be directly evaluated is the result of a dominating misconception in Computer Science, i.e., the view that objects have fixed attributes and inherent qualities that can be asserted on the basis of feature descriptions. The attempt to evaluate designs directly is a form of alienation that strips interactive systems from their usage contexts, attributing quality in use to artefacts, rather than to the interaction of real humans with their own technologies. Interaction in turn must not be evaluated as a thing-in-itself, as is the case with quality in use approaches. Nor must fit to context be seen as the end point of successful design. Instead, the aim of design is to create new forms of value, and it is the achievement of value in the world that we should be evaluating.

We therefore need to develop value-centred frameworks for interactive systems development. These require three novel development products, with associated activities:

- Statements of intended value
- Value delivery scenarios
- Value impact assessment

The move from existing development methodologies to a value-centred one is thus dependent on our ability to:

- devise formats for intended value statements
- author effective value delivery scenarios
- assess the impact of actual and predicted user difficulties and contextual misfit on achieved value

Value-centred development creates further challenges in separating evaluation from iteration. This highlights the limitations of existing usability engineering approaches to causal analysis and design change recommendation. These tend to get buried in the corners of existing evaluation methods, but once they are isolated and scrutinised, there is little of substance to them. By restricting evaluation to the assessment of achieved value, separate iteration activities are required to bring all development resources to bear on well grounded and broadly based design change recommendations. A clean break is needed from past muddling through, and identifying iteration as a distinct process in its own right allows this. A new research area is needed to establish effective and credible iteration methods.

The belief that we can improve some direct interplay between design and evaluation is a logical consequence of both confusing the processes of evaluation and iteration, and also of seeing evaluation as the direct assessment of systems rather than an analysis of the consequences of adverse interactions for the achieved value of a digital product or service. Once we realise that we must separate iteration from evaluation, and that we must evaluate, not systems, but interactions, and evaluate on the basis of achieved value, then we are clearly directed to value-centred development frameworks that are grounded on value enabling interactions rather than on the creation of inherently and intrinsically usable artefacts.

Value-centred HCI is at a very early stage. Much work needs to be done to move it from a set of arguments and potential approaches to a set of proven development approaches. However, it already has value (!) as a conceptual framework that reframes and clarifies several key issues in HCI.

6 REFERENCES

[1] Barnum, C., Bevan, N., Cockton, G., Nielsen, J., Spool, J., and Wixon, D., "The "Magic Number 5": Is It Enough for Web Testing?", in *CHI 2003 Extended Abstracts*, eds. G. Cockton *et al.*, 698-699. ACM Press: New York, 2003.

[2] Beyer, H. and Holtzblatt, K., *Contextual Design*, Morgan Kaufman, 1998.

[3] Cockton, G., "From Quality in Use to Value in the World", in *CHI 2004 Extended Abstracts*, ACM Press: New York, ISBN: 1-5811 3-703-6, 1287-90, 2004.

[4] Cockton, G., "Three and a Half Decades of HCI: Three Bricks Walls and Half a Ladder," in *Proceedings of HCI 2004*, Volume 2, eds. A. Dearden and Leon Watts, Research Press International, 17-20, ISBN:1-897851-13-8, 2004.

[5] Cockton, G., "Value-Centred HCI", Proceedings of the Third Nordic Conference on Human-Computer Interaction, ed. A. Hyrskykari, 149-160, ISBN 1-58113-857-1, 2004.

[6] Cockton, G. and Lavery, D. "A Framework for Usability Problem Extraction", in *INTERACT 99 Proceedings*, eds. A. Sasse and C. Johnson, 347-355, 1999.

[7] Cooper, A. and Reimann, R.M, *About Face 2.0: The Essentials of Interaction Design*, Wiley, 2003.

[8] Felluga, D., "Modules on Marx: On Fetishism." Introductory Guide to Critical Theory. http://www.purdue.edu/guidetotheory/marxism/modules/marxfetishism.html, Purdue University, last update: 28/11/03, last visited 8/10/04.

[9] Freesearch dictionary, http://www.freesearch.co.uk/dictionary/evaluate, last visited 8/10/04.

[10] Gram, C. and Cockton, G. (co-editors of monograph). Design Principles for Interactive Systems, Chapman and Hall, 248 pages, ISBN 0 412 72470 7, 1996.

[11] Marx, K., *Economic & Philosophical Manuscripts of 1844*, Progress Publishers, Moscow 1959.

[12] D. Thompson (ed.), *The Concise Oxford English Dictionary*, 9th edition, 1999.

[13] Whiteside, J., Bennett, J., & Holtzblatt, K., "Usability engineering: Our experience and evolution," in *Handbook of Human-Computer Interaction*, 1st edition, ed. M. Helander, North-Holland, 791-817, 1988.

[14] Wilson, C., "Reader's Questions: Severity Scale for Classifying Usability Problems," *Usability Interface*, 5(4), April 1999, available at http://www.stcsig.org/usability/newsletter/9904-severity-scale.html