Comparing Usability Problems and Redesign Proposals as Input to Practical Systems Development

Kasper Hornbæk & Erik Frøkjær
Department of Computing, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen, Denmark
{kash,erikf}@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 present a study of how developers of a large web application assess usability problems and redesign proposals as input to their systems development. Problems and redesign proposals were generated by 43 evaluators using an inspection technique and think aloud testing. Developers assessed redesign proposals to have higher utility in their work than usability problems. In interviews they explained how redesign proposals gave them new ideas for tackling well known problems. Redesign proposals were also seen as constructive and concrete input. Few usability problems were new to developers, but the problems supported prioritizing ongoing development of the application and taking design decisions. No developers, however, wanted to receive only problems or redesigns. We suggest developing and using redesign proposals as an integral part of usability evaluation.

ACM Classification
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

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

INTRODUCTION

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

Techniques for usability evaluation help designers predict how interacting with their designs may cause users problems, and thus what parts of the designs to improve. Usability evaluation techniques include think aloud testing [24,28], where users solve typical tasks with a design while continuously verbalizing their thoughts, and heuristic evaluation [30,32], where the design is assessed with heuristics such as 'speak the user's language' and 'provide shortcuts'. Extensive research has reported case studies on the use of evaluation techniques [16], compared the performance of techniques [9,17,22,36], and led to reviews of what we know (and don't know) about evaluation techniques [2,10,11].

Most of this research assumes that good usability evaluation techniques are those that best support an evaluator in generating problem descriptions while using the techniques; Hartson et al. [11], 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. First, problem descriptions are sometimes very brief. The 46 usability problems described in [19, appendix 1], for example, are on the average about 28 words long. Therefore, problem descriptions may appear unclear or incomprehensible to readers other than the evaluator.

Second, 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 [23]; as an example, are the problems 'The layout of menus are confusing and the user failed to understand any logic underlying it' and 'The user expected to find the Print command in the File menu, and appeared confused when finding it listed under Functions' similar or not, and if so in what sense?

Third, sometimes no design exists that alleviates 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.

Fourth and finally, generation of lists of usability problems may not matter much in practical systems development. Wixon [39] comments on a recurring discussion regarding comparison of evaluation techniques that '[i]t is short

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.

sighted in that it ignores that problems should be fixed and not just found'.

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

Proposals for redesigns are sometimes integrated with the description of usability problems as quick fixes [3,5,23]; no studies have investigated redesign proposals as a distinct and systematic outcome of usability evaluation. In contrast to the limitations listed above, redesign proposals could be easier to understand; be directly integrated into the design, if appropriate; and be more stimulating to developers. However, we do not know if these benefits materialize in practice, nor do we know what elements good redesign proposals contain.

This paper therefore explores the differences between using descriptions of usability problems on the one hand and redesign proposals on the other as inputs to systems development. We do so in an experiment that compares problems and redesigns on a variety of measures, including developers' assessments. We also investigate whether empirical evaluation techniques are more effective in generating useful redesign proposals than usability inspection techniques. The long term aims of this work are to identify evaluation deliverables more valid than the widely used counting and classification of usability problems, and to find outcomes of usability evaluation pertinent to practical systems development.

RELATED WORK

A number of studies have argued that redesign proposals should form part of usability evaluations. Jeffries [18], for example, suggests that problem reports should contain a description of the problem (and a justification why the current situation is a problem), but also a description of the proposed solution and a justification of why it is better. Lavery et al. [23] likewise recommend having a part of problem report that describe a possible solution to the problem, i.e. 'what is your recommended solution to this problem?' (p. 257). The literature mainly aimed at usability practitioners likewise argues that redesign proposals may be usable to developers [34].

In addition, reflections upon limitations of current comparisons of evaluation techniques have also spurred interest in redesign proposals. Smith and Dunckley [37], for example, argues that

> A number of studies have been carried out to compare usability evaluation methods … However all these have focused on evaluation methods themselves rather than on their influence on design. The effectiveness of the different methods has been compared in terms of the usability problems identified with an assumption of a direct link to design improvements. (p. 832)

Cockton et al. [2], in a review of usability inspection techniques, similarly points out that 'Current UIMs [usability inspection methods] provide little, if any, support for the generation of recommendations for fixing designs to avoid predicted problems' (p. 1120). Wixon [39] is even more harsh in arguing that '[t]he literature on usability evaluation is fundamentally flawed by its lack of relevance to applied usability work' (p. 34). He sees the focus on finding, rather than fixing, problems as one of these flaws.

Surprisingly, only a couple of studies have taken up this challenge and investigated redesign proposals as an outcome of usability evaluation [7,21,31,35]. For example, Dutt et al. [7] considers the ability of heuristic evaluation and cognitive walkthrough to produce requirements for redesigns. While requirements are related to a specific technique, the study does not describe the format or nature of those requirements. The study by Sawyer et al. [35] 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 points out particularly useful aspects of redesign proposals.

Other studies have evaluated usability evaluation techniques by implementing redesigns intended to solve the usability problems predicted [1,20]. John and Marks [20], for example, tracked the influence of fixing usability problems on usability by conducting tests of the system which had had the problems attempted corrected. We applaud their effort to do realistic assessment of evaluation techniques, but the study does not describe how the developer of the system made use of the evaluators' insights. Further, redesigns suggestions contained in the evaluators' problem description reports only address the individual problems. The study by Bailey et al. [1] is special in considering the impact of following redesign proposals from evaluators on measurable aspect of usability. The study does not, however, explore specifically whether redesign proposals work better than descriptions focusing on problems.

In practical usability work, redesign proposals are often made in the form of quick fixes. Dumas et al. [5] 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 is called for.

In summary, related work provides some arguments for redesign proposals as (part of) the result of usability evaluation. However, none of the studies 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.

EXPERIMENT

To begin addressing the questions raised above, we present an exploratory study in which developers' assessments are used to compare the utility of problem reports and redesign proposals. Using developers' realistic assessment of usability problems have been suggested in e.g. [11] and used by e.g. [15]; we here extend that assessment to include also redesigns. The rationale for using developers' assessments as data is that, whatever biases they may have, it is normally their choice if and how to alleviate usability problems and to implement redesign proposals.

More specifically we aim to:

(1) Compare quantitatively the assessment of both usability and utility of problems and redesign suggestions,

(2) Collect qualitative data on the aspects of usability problems and redesign proposals that impact developers' assessment of utility, and

(3) Compare whether inspection and empirical evaluation techniques differ in their ability to generate redesign proposals and usability problems.

The overall procedure of the experiment is outlined in Figure 1.

Evaluators

43 undergraduate and graduate students chose to conduct the evaluation and redesign in a class on HCI and systems design. To encourage participation and ensure genuine motivation, evaluators were at all times free to do another assignment instead of the evaluation and redesign.

Application

The evaluators evaluated one of the largest job portals in Denmark, http://www.jobindex.dk. Jobindex has around 230.000 unique visitors each month, placing it among the top 30 of the most visited Danish web sites. To focus the evaluation, the evaluators only considered three key parts of Jobindex: (1) searching for jobs, (2) creating a CV and personal profile, and (3) web pages providing tips and tricks on how to search for jobs. Jobindex had previously used external usability evaluation of their site.

Evaluation techniques

We chose to compare two evaluation techniques because it has been suggested that empirical usability techniques would be more effective in pointing out how to fix problems [12]. In addition, a recent review of user testing

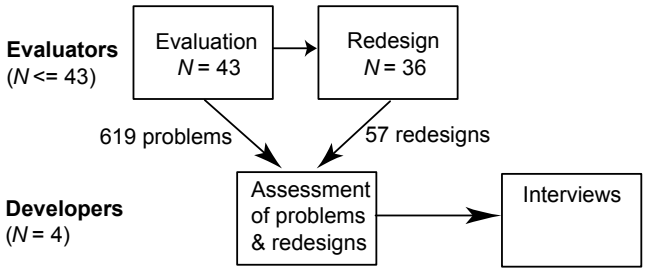

Figure 1: Procedure showing activities for evaluators (top row) and developers (bottom row).

points out that whether 'usability testing uncovers the true problems has not been established' [6, p. 1113]. Developers' assessments of utility of problems may be relevant to further compare user testing and usability inspection. Thus, 21 evaluators received reference [28] as description of think aloud user testing (an empirical usability evaluation technique); twenty-two evaluators received reference [13] as description of the usability inspection technique called metaphors of human thinking (an inspection technique).

Think aloud user testing (TA) is the most popular usability evaluation technique [33,38]. The basic procedure of TA is to give a typical user some realistic tasks to perform with the system under evaluation. While doing the tasks, the user is asked to continuously say out loud what he or she is thinking about. From the user's behavior and talking, the evaluator generates descriptions of usability problems with the interface. For more detail on the procedure see [24,28]

Metaphors of human thinking (MOT) [8,13-15] is a usability inspection technique, in which user interfaces are inspected using metaphors of habit, stream of thought, awareness, utterance, and knowing. These metaphors are intended to break fixed conceptions and help the evaluator focus on users' mental activity during interaction. The reason for choosing this technique is that it has showed promising performance in comparison to popular inspection techniques such as heuristic evaluation [15] and cognitive walkthrough [14], and was liked better by evaluators than cognitive walkthrough [14]. For a detailed discussion of the procedure of MOT, see [13].

Procedure for evaluation

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.

For each usability problem the evaluators identified, they were instructed to give (a) a brief title, (b) a detailed description, and (c) a seriousness rating. Evaluators chose seriousness ratings from a commonly used scale [26, p. 111]: *Rate 1* is given to a critical problem that gives rise to frequent catastrophes which should be corrected before the system is put into use. This grade is for those few problems

that are so serious that the user is better served by a delay in the delivery of the system; *Rate 2* is given to a serious problem that occasionally gives rise to catastrophes which should be corrected in the next version; and *Rate 3* is given to a cosmetic problem that should be corrected sometime when an opportunity arises.

Procedure for redesign

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.

Note that we chose to let evaluators put relatively much work into preparing the redesign proposals. The idea is to provide more details than what is typically done in the cases where usability problems are accompanied by a brief recommendation, as some evaluators quite naturally include when describing usability problems.

Developers' assessment

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.

Note that in this particular development context decisions about design and actual development are intertwined. While larger development projects are likely to have a more strict division of labor, the intertwining of design and development is typical of many smaller projects. Also note that the team did not include a usability specialist.

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.

For each problem or redesign, the developers assessed the following:

- How severe is the problem? The severity of the problem related to users' ability to do their tasks was judged as 1 (very critical problem), 2 (serious problem), 3 (cosmetic problem), or NA (not a problem). Note that this grading is different from the evaluators' seriousness ratings in that only the nature of the problem is being assessed, not when the problem should be corrected which is contingent upon resources within the development organization.
- How frequent will users encounter the problem? Here we used a three grade answer used in previous comparisons of usability evaluation techniques [27]: (1) often, (2) sometimes, or (3) rarely.
- How persistent is the problem? Again we used as inspiration previous work [27] to create the following scale: (1) long, i.e. the problem will continue to bother the user and demand special attention; (2) medium, i.e. the problem eventually disappears, but only after having bothered the user several times; (3) short, i.e. the problem disappears for the user after having been experienced once or twice.
- How useful is the problem in the further development of Jobindex? Does the description of the usability problem or redesign suggestion contain something valuable that you want to use in the future development of Jobindex, for example if you find something new or get ideas for improvements. To answer this question, the developer put a cross on a continuous/graphical rating scale (shown as a 100 mm horizontal line) with the end points labeled 'not useful' and 'very useful'. With this measure, we specifically want to avoid developers feeling constrained by a small number of grades or categories. We quantify utility as the number of millimeters from the 'not useful' end point to the place where the developer had put a cross.

In total, developers reported having spent around 40 hours to assess the problems and redesigns, excluding the time used for the interviews described next.

Interview with developers

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. At the end of the interview, we asked the developers about their understanding of the rating scales they had used. We

also presented them with preliminary results from the study in order to hear their interpretation. 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.

RESULTS

We first go through some data on the evaluation performance, then characterize developers' assessment of problems and redesigns, and finally present qualitative data from interviews with the developers.

Evaluators' performance

Table 1 summarizes the evaluators' performance. In all, 619 problems were identified.

No significant difference were found between techniques in the number of problems evaluators identified, $F(1, 41) = 0.2$, $p > .8$. Evaluators on average identified between 14 and 15 problems. Individual differences were large; some evaluators identified more than 30 problems, others identified less than five.

Evaluators rated problems identified by use of MOT slightly less severe ($M = 2.45$) than problems identified while using TA ($M = 2.37$). Especially it seems that MOT identifies fewer problems graded as serious than does TA, (MOT 35%; TA 44%). These differences, however, are not significant, $F(1, 41) = 0.89$, $p > .3$.

Table 1: Evaluators' performance

	TA (N=21)	MOT (N=22)
Number of problems	14.6 (*SD* = 7.6)	14.2 (*SD* = 9.3)
Seriousness (avg.)	2.37 (*SD* = 0.93)	2.45 (*SD* = 1.20)
Critical problem	9%	10%
Serious problem	44%	35%
Cosmetic problem	47%	55%

Developers' grading of problems and redesigns

Table 2 shows the average of the developers' grading of problems and redesigns. Below we analyze the data using multivariate analysis of variance on the developers' assessments.

Overall, we see no differences between techniques in the assessment of problems and redesigns, $F(4,599) = 0.31$, $p > .5$. As Table 2 shows, the assessment of problems is numerically quite similar for severity, frequency, and persistence. Problems produced by the two techniques also seem to be valued as having equal utility in the development process. When rating severity, developers could also assess the problem as not being a usability problem. The number of such assessments was low, for both techniques about 7% of the problems. Similarly, redesigns are not assessed significantly differently between techniques.

Of particular interest here is the difference between problems and redesigns. We find an overall significance in the developers' assessment of these, $F(4, 599) = 14.63$, $p < .001$. The sources for the differences between the assessment of problems and redesigns appear to be three. Developers consider problems underlying redesigns more frequent than those described in the problem descriptions; redesigns 2.32 ($SD = 0.35$), problems 2.44 ($SD = 0.42$), $F(1, 602) = 4.36$, $p < .05$. Developers also consider problems underlying redesigns more persistent than those described in the problem descriptions; redesigns 2.32 ($SD = 0.35$), problems 2.44 ($SD = 0.42$), $F(1, 602) = 6.96$, $p < .01$. As a concrete illustration of this last observation, developers assess 56% of the problems to be of short persistence (i.e. the problem disappears after bothering the user once or twice). Only 38% of the problems underlying the redesigns are considered of short persistence.

The largest difference seen by developers between problems and redesign proposals appear to be their utility in redesign, $F(1, 602) = 57.37$, $p < .001$. Redesigns are seen as more useful ($M = 38.6$, $SD = 18.0$) than problem descriptions ($M = 25.7$, $SD = 11.3$). One way of illustrating

Table 2: Developers' assessment of usability problems and redesigns.

Question	Usability problems		Redesigns	
	TA (*N* = 321)	MOT (*N* = 298)	TA (*N* = 28)	MOT (*N* = 29)
How severe is the problem described or attempted solved? (1 = very critical, 2 = serious, 3 = cosmetic)	2.64 (0.43)	2.70 (0.38)	2.41 (0.36)	2.39 (0.32)
How frequent will users encounter the problem? (1 = often, 2 = sometimes, 3 = rarely)	2.42 (0.41)	2.46 (0.42)	2.38 (0.35)	2.28 (0.34)
How persistent is the problem described or attempted solved? (1 = long, 2 = medium, 3 = short)	2.45 (0.54)	2.46 (0.52)	2.23 (0.48)	2.24 (0.46)
How useful is the problem in the further development of Jobindex? (graphical rating scale from 1, not useful, to 100, very useful)	25.9 (10.7)	25.4 (11.9)	37.1 (16.9)	40.0 (19.2)

this difference is the observation that 77% of the redesigns are assessed to be of higher utility than the average problem. Only 15% of the problems are considered of higher utility than the average redesign.

On the surface, the differences in utility between problem descriptions and redesign proposals could be a consequence of evaluators choosing problems to redesign that they considered particularly serious to users. This, however, is not a valid explanation, as can be illustrated by the evaluators' own pointing out of the problems underlying their redesign proposals. In the redesigns most evaluators made cross references to numbers on their lists of usability problems, indicating a total of 117 problems on which their redesigns were based. However, the developers' assessment of utility were not significantly different between those problems attempted solved in a redesign (M = 27.9, SD = 11.5) and those not solved (M = 25.1, SD = 11.2), $F(1, 552) = 3.06$, $p > .05$. The difference in utility between redesigns and problems thus seems to stem from some aspect of the redesign proposals.

Interviews with developers

To gain further insights into the utility of usability problems and redesign proposals, we systematically analyzed the interviews with developers. We extracted from the interviews all statements about qualities of either usability problems or redesigns, together with statements about use of problems and redesigns as input to development. Below we present this data; tables 3 and 4 give a summary.

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 therefore 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 ad] 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

Table 3: Characteristics of usability problems as discussed by the developers, excluding the web content manager. N refers to the number of developers mentioning the characteristic.

Characteristic	N
Mostly already known problems	3
Supports ongoing design discussions and decisions	3
Help prioritizing what parts of the UI to address	3
Sometimes lack context for problem and convincing arguments for users' difficulties	2
Convincing in referring to users	1

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.

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 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 county … 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]

Table 4: Characteristics of redesign proposals as seen by the developers, excluding the web content manager. N refers to the number of developers mentioning the characteristic.

Characteristic	N
Gives ideas, especially on hard well-known problems	3
Easier and more distinct to remember	3
More concrete, e.g. through drawings and code fragments	2
More coherent and worked through	2

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 was 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 although 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'.

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.

DISCUSSION

The study shows that developers value redesign proposals as input to their development work. The assessments of the utility of redesign proposals are higher than those of usability problems overall, and also higher than the usability problems that the redesigns aimed at alleviating. This supports the first hypothesis of the experiment. 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 to form part of the input from usability evaluation to systems development.

These results suggests that usability evaluations should place more focus on developing and reporting such proposals than is typically done, cf. the section Related Work.

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 [39] 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 [10,14]. 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.

A likely objection to this study is that we are comparing apples and pears, in that large differences exist between descriptions of usability problems and redesign proposals, for example in length, layout, and work effort invested in each of them. Our answer is simply that in this study the different characteristics of each kind of input seem to be valuable to developers. However, the present study is only a first step towards characterizing what elements of redesigns that developers find to be of utility in their work. The concrete, illustrated, and carefully prepared redesign proposals aimed at in this study are quite different from the quick fixes included in some usability reports—we do not know how they compare to our redesign proposals.

Similarly to [5], our study also suggests how to describe usability problems in a way so that they are useful to developers. From the interviews, it was clear that developers occasionally missed a clear rationale for a usability problem or a convincing argument for the expected impacts of a usability problem.

Another aim of the study was to identify differences between inspection techniques and empirical usability evaluation techniques. In this study, however, think aloud and metaphors of human thinking perform equally well. We find no evidence that either leads to usability problems or redesigns that are assessed differently by developers. This is somewhat surprising for two reasons. First, previous work suggests that think aloud testing would be superior in pointing out useful redesign proposals, by providing 'genuine and applicable feedback to system designers' [12, p. 506]. Second, at least one developer expressed a great reliance upon and trust in problems that explicitly mention test users. Many such problems were among the ones he assessed, yet no difference between techniques was found for this developer or for the others. Possibly, there is some quality of problems produced with metaphors of human thinking, and perhaps with inspection techniques in general, that we can not describe accurately at the moment.

In the literature, several arguments against emphasizing redesign proposals have been put forward. Mack and Montaniz [25], for example, remarks that:

> There is seldom only one solution to a problem and solving a problem has costs in a larger development context. For example, we may consider solving a

> problem by modifying the interface, elaborating on training or online help, or we may decide that the benefit of the tool outweighs the potential problems and user dissatisfaction (p. 337)

Our data, however, did not indicate that developers were put off by a particular choice in a redesign. One of them, for example, remarked that he found the idea of a particular redesign proposal to be of quite high utility, although he would do it differently than proposed.

It has also been suggested that 'evaluators often go immediately to solutions, without describing the problems that need to be solved' [18, p. 277]. Such practice might produce interesting redesigns, but could lead to less emphasis on the user and the users' tasks. Similarly, Doubleday et al. [4] warns that 'not understanding the underlying cause has implications for re-design as a new design may remove the original symptom but if the underlying cause remains, a different symptom may be triggered' (p. 109). From the assessments and interviews, it is difficult to reach any firm conclusions on whether redesign proposals lead developers to lose focus on the users. In some interviews, developers related redesign proposals to details of the users work and the organizational context in which the application was used; in a few others, they quickly presented technical arguments for or against redesigns.

Reflecting more broadly on the study, two comments are pertinent. First, the interviews illustrate how developers emphasize expressions and opinions that originate from the users. Future work in how we present problems, redesigns and other results of user interaction with designs should take this into consideration; for example, we known of no attempt to let users rate aspects of usability problems such as severity or frequency.

Second, it is interesting that developers find the problems identified to be mainly confirmations of issues they already know. In a comparative usability evaluation, Molich et al. [29] similarly found that only 4% of the problems identified were new to the usability team responsible for the system evaluated. One immediate reaction could be that this is not much. Yet, maybe we should be careful in concluding that developers get few new insights from usability evaluations. The developers in our study actually used both redesigns and usability problems, and their thinking about the application seemed to have been influenced. Further, developers who for years have worked intensively with the application and its use context will not easily take up results of usability evaluations. On the contrary, changing their understanding is a process requiring time, during which new insights does not appear as something distinct and immediately clear. Rather, developers will experience nagging doubts, small changes in thinking, and challenges to their habitual understanding. Studying how this develops over time would probably give a more valid picture of the impact of usability evaluations.

CONCLUSION

So far work on usability evaluation techniques has focused on the techniques' ability to generate problems; proposals for redesigns that alleviate those problems have largely been ignored as an integrated part of usability evaluation. Redesign proposals could, however, be of more practical benefit than usability problems by being more concrete, easier to understand for developers, and more useful in systems development.

This study showed how redesign proposals were assessed by developers as being of higher utility than problem descriptions; even compared to the problems that the redesign aimed to alleviate. Interviews showed that developers appreciated getting ideas from redesigns, and liked the concrete and constructive descriptions. Usability problems were seen more as a help in prioritizing and supporting ongoing design decisions, although developers were already aware of most of the problems. Nevertheless, all developers wanted as input both redesign proposals and usability problems.

The results indicate that redesigns, created during or immediately after usability evaluation, are a useful supplement to descriptions of usability problems. In research on usability evaluation techniques, redesigns comprise an important quality of a technique that should be further investigated and considered in comparisons of techniques. In usability evaluation practice, even if the redesigns are sketchy and do not fit with the overall design and use context of the application, providing redesigns is appreciated and useful to developers.

ACKNOWLEDGEMENTS

We thank the development team at jobindex.dk and the student evaluators for participating in the study. Peter Naur provided helpful comments on a draft.

REFERENCES

1. Bailey, R. W., Allan, R. W., & Raiello, P. Usability Testing Vs. Heuristic Evaluation: a Head-to-Head Comparison, *Proc. Human Factors Society 36th Annual Meeting*, (1992), 409-413.
2. Cockton G., Lavery, D., & Woolrych, A. Inspection-Based Evaluations, in Jacko, J. A. & Sears, A. *The Human-Computer Interaction Handbook,* Lawrence Erlbaum Associates, 2003, 1118-1138.
3. Cockton, G., Woolrych, A., Hall, L., & Hidemarch, M. Changing Analysts' Tunes: The Surprising Impact of a New Instrument for Usability Inspection Method Assessment, *Proc. HCI 2003*, Springer Verlag (2003), 145-162.
4. Doubleday, A., Ryan, A., & Sutcliffe, A. A Comparison of Usability Techniques for Evaluating Design, *Proc. DIS'97*, ACM Press (1997), 101-110.
5. Dumas, J., Molich, R., & Jefferies, R. Describing Usability Problems: Are We Sending the Right Message?, *interactions*, 4 (2004), 24-29.

6. Dumas J. User-Based Evaluations, in Jacko, J. A. & Sears, A. *The Human-Computer Interaction Handbook,* Lawrence Erlbaum Associates, 2003, 1093-1117.
7. Dutt, A., Johnson, H., & Johnson, P. Evaluating Evaluation Methods, *Proc. HCI 1994*, Cambridge University Press (1994), 109-121.
8. Frøkjær, E. & Hornbæk, K. Metaphors of Human Thinking in HCI: Habit, Stream of Thought, Awareness, Utterance, and Knowing, *Proc. HF/OzCHI 2002* (2002).
9. Fu, L. & Salvendy, G. Effectiveness of User-Testing and Heuristic Evaluation as a Function of Performance Classification, *Behaviour and Information Technology*, *21*, 2 (2002), 137-143.
10. 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.
11. 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.
12. Helms Jørgensen, A. Thinking-Aloud in User Interface Design: a Method Promoting Cognitive Ergonomics, *Ergonomics*, *33*, 4 (1990), 501-507.
13. Hornbæk, K. & Frøkjær, E. Evaluating User Interfaces with Metaphors of Human Thinking, *Lecture Notes in Computer Science 2615*, Springer (2002), 486-507.
14. Hornbæk, K. & Frøkjær, E. Two Psychology-Based Usability Inspection Techniques Studied in a Diary Experiment , *Proc. NordiCHI 2004*, ACM Press (2004)
15. Hornbæk, K. & Frøkjær, E. Usability Inspection by Metaphors of Human Thinking Compared to Heuristic Evaluation, *International Journal of Human-Computer Interaction*, *17*, 3 (2004), 357-374.
16. Jacobsen, N. E. & John, B. E. Two Case Studies in Using Cognitive Walkthroughs for Interface Evaluation, *CMU-CS-00-132* (2000).
17. Jeffries, R., Miller, J., Wharton, C., & Uyeda, K. User Interface Evaluation in the Real World: A Comparison of Four Techniques, *Proc. CHI'91*, (1991), 119-124.
18. Jeffries R., Usability Problem Reports: Helping Evaluators Communicate Effectively With Developers, in Nielsen, J. & Mack, R. L. *Usability Inspection Methods,* John Wiley, 1994, 273-294.
19. John, B. E. & 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).
20. John, B. E. & Marks, S. J. Tracking the Effectiveness of Usability Evaluation Methods, *Behaviour and Information Technology*, *16*, 4/5 (1997), 188-202.
21. Johnson H., Generating User Requirements From Discount Usability Evaluations, in Harris, D. *Engineering Psychology and Cognitive Ergonomics, Vol. 2,* Ashgate Publishing, 1997, 339-357.
22. Karat, C.-M., Campbell, R., & Fiegel, T. Comparison of Empirical Testing and Walkthrough Methods in Usability Interface Evaluation, *Proc. CHI'92*, ACM Press (1992), 397-404.
23. 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.
24. Lewis, C. Using the "Thinking-Aloud" Method in Cognitive Interface Design, *Research Report RC9265* (1982).
25. Mack R. L. & Montaniz, F., Observing, Predicting, and Analyzing Usability Problems, in Nielsen, J. & Mack, R. L. *Usability Inspection Methods,* John Wiley and sons, 1994, 295-339.
26. Molich, R. *Brugervenlige Edb-Systemer (in Danish)*, Teknisk Forlag, 1994.
27. Molich, Rolf, Comparative Usability Evaluation, 2003, www.dialogdesign.dk/cue.html.
28. Molich, Rolf, User testing, Discount user testing, 2003, www.dialogdesign.dk.
29. Molich, R., Ede, M. R., Kaasgaard, K., & Karyukin, B. Comparative Usability Evaluation, *Behaviour and Information Technology*, *23*, 1 (2004), 65-74.
30. Molich, R. & Nielsen, J. Improving a Human-Computer Dialogue, *Communications of the ACM*, *33*, 3 (1990), 338-348.
31. 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.
32. Nielsen, J. & Molich, R. Heuristic Evaluation of User Interfaces, *Proc. CHI'90*, ACM Press (1990), 249-256.
33. Rosenbaum, S., Rohn, J., & Humberg, J. A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys, *Proc. CHI 2000*, ACM Press (2000), 337-344.
34. Rubin, J. *Handbook of Usability Testing*, John Wiley & Sons, New York, NY, 1994.
35. Sawyer, P., Flanders, A., & Wixon, D. Making a Difference - The Impact of Inspections, *Proc. CHI'96*, ACM Press (1996), 376-382.
36. Sears, A. & Hess, D. Cognitive Walkthroughs: Understanding the Effect of Task Description Detail on Evaluator Performance, *International Journal of Human-Computer Interaction*, *11* (1999), 185-200.
37. Smith, A. & Dunckley, L. Prototype Evaluation and Redesign: Structuring the Design Space Through Contextual Techniques, *Interacting with Computers*, *14* (2002), 821-843.
38. Vredenburg, K., Mao, J.-Y., Smith, P. W., & Carey, T. A Survey of User-Centered Design Practice, *Proc. CHI 2002*, ACM Press (2002), 472-478.
39. Wixon, D. Evaluating Usability Methods: Why the Current Literature Fails the Practitioner, *interactions*, *10*, 4 (2003), 29-34.