Datalogisk Institut
KØBENHAVNS UNIVERSITET

DIKU

What Makes a Developer's Heart Tick? Characterizing Effective Feedback from Usability Evaluation

Mie Nørgaard & Kasper Hornbæk

Technical Report no. 07/01
ISSN: 0107-8283

Dept. of Computer Science
University of Copenhagen • Universitetsparken 1
DK-2100 Copenhagen • Denmark

Running head: What Makes a Developer's Heart Tick?

What Makes a Developer's Heart Tick?

Characterizing Effective Feedback from Usability Evaluation

Mie Nørgaard & Kasper Hornbæk

University of Copenhagen, Denmark

December 2006

Abstract

The format used to present feedback from usability evaluations to developers affects whether problems are understood, accepted and fixed. Yet, little research has investigated which formats are the most effective. We describe an explorative study where three developers assess 40 usability findings presented using five feedback formats. Data suggest that feedback serve multiple purposes. Initially feedback must be convincing and convey an understanding of the problem. Next, feedback must be easy to use before merely serving as a reminder of the problem. Prior to working with the feedback developers rate redesign proposals, multimedia reports, and annotated screendumps as more valuable than lists of problems, which again are rated more valuable than a scenario type format. After working with the feedback, developers rate the value of formats alike. This reflects how all formats may serve to remind but that redesign proposals, multimedia reports, and annotated screendumps best address feedback's initial purpose.

1 Introduction

Since usability studies became established as a important activity in systems development the effectiveness of usability evaluation methods has been investigated thoroughly (see for instance (Jeffries, Miller, Wharton, & Uyeda, 1991; John & Marks, 1997; Sears, 1997)). The literature reveals a strong focus on comparing usability evaluation methods but how the evaluation results are fed back to a design team has not been the focus of much work, but see (Dumas, Molich, & Jeffries, 2004; Hornbæk & Frøkjær, 2005). This is unfortunate since one goal of usability evaluation is to improve systems. To reach this goal evaluations must move beyond solely listing usability problems (UPs) and help developers to decide which UPs to fix and to understand how to can fix them.

In one English dictionary (askoxford.com) feedback is described as: ‘Information given in response to a product, performance etc., used as a basis for improvement’. According to this definition feedback needs to fulfill certain requirements to be successful: the receiver needs to understand the feedback and the feedback needs to facilitate a solution to a given problem. To do this, the feedback needs to be convincing. Consequently there are at least two challenges for an evaluator about to feed back results to a development team: First, developers may not be easily convinced about usability problems, either believing that the system is great as it is, or that users eventually will come around to using it (Kennedy, 1989; Seffah & Andreevskaia, 2003). Second, developers might not be hostile to changes, but simply find it difficult to understand a UP because it is vaguely described (Dumas et al., 2004). How evaluators tackle these two challenges can influence the evaluation’s impact dramatically.

The present explorative study aims at contributing to our understanding of the practical use of different feedback formats and thus identify how we more successfully can feed usability findings back to developers. The study investigates how five feedback formats that represent different ways an evaluator may deliver usability results, are used and assessed by developers. The results suggest that developers initially value information in addition to a problem descripton, such as videohiglights, contextual screendumps and redesign proposals, but after having worked with the feedback the differences between feedback formats diminish. We argue that these results are important for usability practitioners as advise for chosing between feedback formats and for reasearchers as a help to understand the roles of feedback.

2 Related work

Related work can be divided into two categories; one characterizing feedback practices and another concerned with feedback research. Below we discuss the two categories in turn.

2.1 Feedback practices

To facilitate the improvement of a system, feedback from usability evaluations often include descriptions of a problem’s severity (Kennedy, 1989; Dumas, 1989; Coble, Karat, & Kahn, 1997; Hornbæk & Frøkjær, 2005), the context of the problem (Kennedy, 1989; Nayak, Mrazek, & Smith, 1995), redesign proposals (Jeffries, 1993; Nayak et al., 1995; Dumas et al., 2004; Hornbæk & Frøkjær, 2005), and underlying causes of problems (Dumas, 1989). Practitioners and researchers also agree on the persuasive power of developers seeing users interact with the system (Schell, 1986; Mills, 1987; Dumas, 1989; Redish et al., 2002). Below we describe different approaches to providing feedback.

An informal survey on an online forum for usability pratitioners suggests that a usability report focussing on a *list of problems* is perhaps the most common way to feed back usability results to developers. The importance of presenting positive comments together with the UPs is often argued (Dumas et al., 2004). In a problem list each UP may be described by a short text and a severity rating; these ratings may be used to present a *top 10-list* of the most critical problems to help developers prioritize their work and cut down on the number of problems reported (Dumas, 1989; Nielsen, 1993; Nayak et al., 1995; Redish et al., 2002).

A *GUI binder* is described as a collection of screendumps annotated with recommended usability enhancements (Nayak et al., 1995). This feedback format aims at providing developers with example-based references to support the development process.

Nayak et al. (Nayak et al., 1995) describe *multimedia presentations* as interactive documents that mixes desciptive text with videohighligts, pictures and graphics. The information is linked in a structure similar to web pages.

As an elaboration of the video highlights used for multimedia presentations, Dumas and Redish discuss a *professional video production* that resembles video productions as we know them from TV, including a narrator, voiceover and examples from the test (Dumas & Redish, 1999).

Redesign proposals are referred to as constructive input, that provides developers with ideas for tackling problems (Hornbæk & Frøkjær, 2005). Redesign proposals can include a brief summary of the redesign, a justification of the proposed design, an explanation of the interaction and design decisions in the redesign and finally illustrations of how the redesign works (Jeffries, 1993; Hornbæk & Frøkjær, 2005).

Scenarios can be defined as ‘a succinct story describing a user’s goal, start point, and intermediary factors that relate to product use’ (Kahn & Prail, 1993). They build upon results from real users (Nayak et al., 1995) and task analysis (Nielsen, 1993). Scenarios are only rarely mentioned as a way to provide feedback. This surprise, since their strength is portraying context of use and user behaviour, which are important when understanding a problem. *Human-centered stories* are one type of scenario. In style they resemble fiction writing using dialouge and describing the characters’ emotions and motivations (Strom, 2003).

The value of *oral feedback* as a means to describe and initiate a dialogue about results is often mentioned in the literature (Butler & Ehrlich, 1993; Kahn & Prail, 1993; Dumas & Redish, 1999). Face-to-face presentation has the power to clear up potential misunderstandings in an engaging and convincing interaction between evaluator and developer.

2.2 Feedback research

Based on the lack of studies of feedback and the diversity of formats available we need to study how developers assess feedback from usability evaluation. Cockton recently argued that usability studies are moving from looking at evaluations as merely being problem list generators to also dealing with the problems’ impact (Cockton, 2006).

One line of work pointing in this direction is the work concerning downstream utility (John & Marks, 1997; Hornbæk & Frøkjær, 2005; Law, 2006) which concerns the effectiveness with which a solution to a UP is implemented. In a study of downstream utility, Law points to issues such as ‘credibility’ as a key factor for effective feedback (Law, 2006) and describes how developers need to be convinced about for example the evaluator’s expertise before taking the feedback to heart. She suggests that the persuasive power of feedback lies in providing the developers with information about the severity of the usability problem, problem frequency as well as elaborate and accurate problem descriptions. Redesign proposals and an estimated fixing effort are also mentioned to be of importance for good feedback.

At the opening plenary at the Usability Professionals Association Annual Meeting in 1993, Jared Spool from the consultancy User Interface Engineering suggested that usability professionals take a closer look at how they deliver usability feedback, arguing that evaluators should ‘take their own medicine’ when it comes to the usability of their feedback (as referred in (Nielsen, 1994)). A recent special issue also called for more research on the ‘various form of feedback in which the results of usability evaluation is presented to developers’ in order to examine persuasiveness and impact (Hornbæk & Stage, 2006).

This explorative study aims at investigating how such various formats convince developers and provide them with an understanding of usability problems. The short-term goal is to get better knowledge of how evaluators should present their feedback to developers in order for it to be understood and used. The long-term goal is to make evaluation a more powerful player in software development, something only rarely the case today (Hornbæk & Stage, 2006).

3 Method

To identify effective ways of providing feedback, we investigated how five different feedback formats influence usability work in a Danish company. This setup was chosen because it allowed us to study a running system in realistic settings and provided an opportunity to investigate how developers assess feedback when first presented to them, and how they rate the same feedback once they have worked with it.

The explorative study was performed in eight steps. The system was tested, problems ideitified and merged into groups. The problem descripitons were then formatted according to five feedback formats and developers assessed these on five questions. The developers worked with the feedback, assessed it and were finally interviewed about their assessments. The eight steps are described in detail below (see also Figure 1 and 2).

As the five questions show (question 1-5, Table 2) the study was designed to investigate how different feddback formats convinced and provided developers with an understanding of the usability problems. We hypothesised that the first impressions of the feedback and the ratings after use (referred to as pre and post use) would vary, since working intensively with a format might bring the developers to appreciate certain qualities of a format. We also expected the study to provide qualitative data on how to improve feedback from evaluations to developers.

The company in question is Jobindex, a non-hiearchally organised company with 37 employees who provides web based services related to job searching. The three developers who participated in the explorative study comprise the development team concerned with systems development.

3.1 Step one – testing the application

A think aloud test of the system comprised six test sessions and followed the guidelines of (Dumas & Redish, 1999). Jobindex identified the area of interest and approved the tasks for the test. The test sessions were recorded on digital video using a webcam and Tech Smith’s Morae software. The goal of the test was to sample a set of usability findings for the study, not to uncover every issue in the application.

3.2 Step two – analysing the results

To identify UPs, the two evaluators discussed and analysed the test results immediately after each test session, as recommended by (Nørgaard & Hornbæk, 2006). After the six sessions usability findings were consolidated and described with a title, a description of the problem, severity, the context in which the problem occurred and one or more redesign ideas. As recommended by Dumas and Redish (Dumas & Redish, 1993) we included positive findings (PFs). At the end of step two 75 usability findings had been described, comprising 67 UPs and eight PFs.

3.3 Step three - merging usability findings into 40 groups

To eliminate doublets, the 75 usability findings were merged into groups of related problems. The usability findings were merged by rough similarity until 40 groups had emerged. This limit was set to ensure that the developers would get experience in working with each feedback format during step six in the study. Each group consisted of one to six usability findings.

3.4 Step four – turning the findings into feedback items

We chose to investigate five feedback formats that represent different approaches to providing feedback from evaluations. As mentioned above, formats were chosen based on our literaturs review and an informal survey amongst practitioners.

The **list of problems (P)** consists of a description and a severity rating of the UPs. Severity is rated according to a five-step scale (Dumas, 1989). The format is included in the study since it is a common way to present usability feedback that can be produced at low cost. The problem list took approximately half a day to prepare.

The **GUI binder (G)** consists of screendumps annotated with information about where the UP ocurred, a brief description of the UP and a description of one or more possible solutions. The GUI binder is included in the study because it can be produced at a fairly low cost and because it primarily focusses on presenting the context of the problem and only briefly touches upon possible redesign issues. The GUI binder took approximately one day to prepare

The **multimedia presentation (M)** consists of linked html-documents containing a description of the problem, a video with examples of user interaction, a description of one or more solutions, a graphical illustration of severity, illustrative drawings that help to skim the content, illustrations of both problem and possible solution, and finally a short explanation of the illustrations. This format is included in the study because it addresses the recommendations to let developers see real users interact with the system. Also, the multimedia presentation might be more enjoyable to work with since it presents its information in an engaging and varied manner. The multimedia presentation took approximately three days to prepare.

The **redesign proposals (R)** consist of a brief description of the UP, a description of one or more solutions, a justification of the solutions, illustrations of the solutions and finally a short text explaining the illustrations. Redesign proposals are included because justifiations ought to make them a convincing format and the ideas for solutions ought to improve the understanding of the UP and facilitate the actual fixing of the problems. The redesign proposals tok approximately a day and a half to prepare.

Representing scenarios in this study we chose to use **human-centered stories (H).** These are expected to be persuasive and to provide valuable information about the context of use. In this study a human-centered story is approximately one page long and consists of six lines of introduction (presenting the characters and 'setting the stage') and a narrative that describes a problem, the context and the user's motivation and feelings in the situation. The human-centered stories took approximately two days to prepare.

The feedback was presented to the developers on paper (formats PGRH) and CD-rom (format M). We found this most flexible and according to practice. Despite numerous recommendations to interact with developers (Butler & Ehrlich, 1993; Kahn & Prail, 1993; Dumas & Redish, 1999), this explorative study refrains from studying oral feedback. This is not to underestimate the high value of oral feedback, but to emphasize the importance of the deliverables that support the oral feedback and serve as documentation and a reminder for developers during their work.

3.4.1 Producing comparable feedback items

The five formats PGMRH comprise a combination of different descriptive elements such as text, illustrations and severity ratings. We produced a series of descriptive elements to be copy-pasted when we constructed the feedback according to the five formats. We did this to improve the comparability between the formats. For example the same rating would be used for all formats using severity ratings. Step four resulted in a total of 200 so-called feedback items, comprising 35 UPs and 5 PFs described by five feedback formats (see Figure 1).

3.5 Step five – Pre use rating of feedback items

In order to rate the value of the feedback items the 200 items were presented to three developers at Jobindex who usually receive and take care of usability feedback. A description of the test set-up, the participants and the tasks were also provided.

The 200 items were presented in random order so that no one feedback format was favoured by being presented first. Each feedback item was presented with a rating sheet where each developer individually would assess every feedback item according to the questions in Table 2. The questions were intended to shed light on issues such as usefulness, persuasive power and clarity; issues that are crucial for the feedback’s quality. To answer the questions, the developer would mark a point on a 100 mm horizontal line. Each end of the line was marked with the labels shown in parenthesis after the questions (e.g., ‘very poorly’/’very well’). This method of measuring is inspired by (Frøkjær & Hornbæk, 2005) and let the developers answer the questions without being constrained by a small number of categories on the scale. The scale is quantified by measuring the millimetres from the start point to the point on the line marked by the developer. Each developer used approximately four hours rating the feedback items.

3.6 Step six – putting the feedback items into action

After developers had rated their first impressions of the feedback we wanted to study how they would use the feedback in their daily work. Each developer received a set of the 40 usability findings; 32 usability findings in print (covering equally feedback formats PGRH) and the remaining eight usability findings on a CD-rom (M). The feedback items were selected at random from the set of 200 feedback items produced in step four.

The developers were instructed to carry out their work on the system as if they had received any other usability report. This was done so the developers could familiarize themselves with and perhaps change their opinions about some of the feedback items once they got experienced using them. The developers worked with the feedback items for approximately 12 weeks inbetween their other tasks at Jobindex.

3.7 Step seven – Post use rating of feedback items

Having finished the work on the system the developers repeated step five, this time rating just the 40 feedback items they had been working with, keeping their actual work experiences in mind. Each developer used approximately one and a half hours on this task.

3.8 Step eight – individual interviews

Finally, developers were interviewed individually. They were presented with and asked to discuss examples of the feedback items they had rated highest and lowest. They were also asked to discuss the significance of positive findings. Finally they were asked to perform a card sorting exercise in which they discussed the value of feedback elements such as severity ratings, video and contextual screendumps. The aim of the interviews was to get finer nuances on the developers' opinions, collect anecdotal data about their experiences with the feedback formats, and collect ideas for improving feedback on usability evaluation. During the interviews, points and opinions were captured directly on the relevant feedback items. The interviews were afterwards documented with thorough notes, two of the interviews were additionally audio recorded.

4 Results

4.1 Pre use rating

Table 1 presents an overview of developers' mean pre use ratings of the five feedback formats. To protect the experiment-wide error, we first analyzed the pre use ratings using analysis of variance (ANOVA). This test suggests significant differences between feedback formats (see Table 1). Post hoc tests point to redesign proposals, the multimedia presentation and the GUI binder as being rated equal and significantly better than the problem list, which in turn is rated better than human-centered stories. To illustrate this developers rate redesign proposals highest in 40% of the cases, multimedia presentation in 31% of the cases, GUI binder in 23% of the cases, and the problem list in 6% of the cases. Human-centered stories were never rated highest.

To investigate these differences we conducted individual analysis of variance on each question. Table 2 shows how the significant groups change between questions. Of the three top-rated formats (GMR), R is rated significantly higher than M on a question concerning whether a feedback item helps the developer solve the problem. M seems on the other hand to be slightly better at convincing the developer about the problem. The difference to G and R is not significant though. Despite the small variances on questions the ratings generally support the picture from Table 1 of the GUI binder, the multimedia presentation and redesign proposals as being the most valued feedback formats.

We found no significant correlation between order of presentation and ratings, *F(7,167) = 0.54, p<.921*, suggesting that having seen other presentations of a UP does not affect how a developer rates a feedback item.

4.2 Post use ratings

Table 1 also shows the mean ratings developers made of the five feedback formats after having worked with them for three months. An overall MANOVA shows that there are no significant differences in how the five formats are rated after use. An analysis of the ratings of each specific question confirmed the result.

A comparison of the ratings given to identical feedback items pre and post use (Figure 3) show that all five questions receive lower ratings in the post use rating. The only exception is human-centered stories (H), which generally receive the same rating.

4.3 Interviews

We analyzed and consolidated the notes from the interviews into 14 groups of similar opinions. Four of these identified general parameters that make feedback useful to developers; the problem can be recognized, the problem is easy to fix, the feedback contains much information about the problem's context and the feedback is quick and easy to use. Ten groups concerned the feedback formats (Table 4).

4.3.1 General findings – explaining high and low ratings

The interviews showed that the top rated feedback items had some general characteristics in common. First, the problems were recognizable to the developer, meaning that the developer knew about them already. As an example developer 3 (Dev3) explains: 'This is a much more recognizable problem. I know it is annoying. It is a problem I have been in contact with before'. Second, the problems that received high ratings were considered easy to fix: 'It's a change that can be easily overcome…that's why it has a higher rating' (Dev3). Six out of ten high rated feedback items were explained with the fact that developers agreed with the problem. Five of ten high rated feedback formats were explained with problems being easy to fix.

The lowest rated feedback items also showed similarities. A low rated feedback item often described a problem that was hard to recognize either because the developer was not convinced about the problem, or because he needed more contextual information to understand it. Dev2 points to one reason for not being convinced and wanting more context about a problem: 'I am not able to deduct the cause of the problem from this feedback'. Five of ten low rated feedback items were explained with the fact that the developer disagreed with the problem or found it impossible to solve. Developers explained four of ten low rated items with not being able to understand the problem for example; 'I have trouble understanding what it is…I mean what search words the user typed…I understand that the user has typed something and has an expectation about finding something…but I have a hard time understanding what it is' (Dev1).

Generally developers value the access to contextual information and several formats are criticized for not describing enough context. 'I need to know more' Dev 1 points out when discussing several low rated feedback items. Conversely, formats heavy on context are not without problems. Feedback formats, which elaborate on context of use are either criticized for being tedious to use (M) or rated poorly throughout the explorative study (H). This suggests that developers consider the format's ease of use an important parameter when assessing how a format performs.

4.3.2 Details on the five formats

Developers criticize human-centered stories for being time consuming and 'full of noise', as Dev1 puts it. Dev3 joins the criticism with the view that H does not really help to fix the problem and that it is often difficult to understand what the problem is. 'It does surprise that you can still be unsure of what the problem is after having read the long text', he explains. On the positive side H 'shows you where you loose the user' (Dev1) and provide contextual information about the UP, which help to understand the problem (Dev2).

Problem lists are considered fully sufficient for presenting uncontroversial UPs, and Dev2 describes how he uses the severity rating to estimate whether he is 'on the same level as the evaluator'. This is an important part of convincing him about the nature of the problem. Dev3 criticize P for lacking contextual information: 'The problem has been boiled down to one line of text. It can be difficult to understand [the problem] because the description is too short and is does not include any description of context, suggestions for solutions or anything. I often catch myself thinking ‘what am I supposed to do with this?’

The multimedia presentation is mostly valued for the videos by Dev2 who explains how videos provide fine nuances about the context and the use of the system. Dev1 and Dev3 on the other hand values the possibility to dig into a video, but find that the UPs are generally easily understood without seeing the video. They find that describing simple UPs with video unnecessary and criticize M for being too time consuming because of the videos. Dev1 explained how he found the video in M tedious because it was difficult to get a quick overview and to skim the content: ‘I cannot fast forward to the point of the video’ he criticizes. He suggests providing a textual description of the video’s story line, using H as a model. Dev3 supports this idea. The developers do not find that graphical illustrations add any value and call for more thoroughly explained severity ratings.

Dev1 comments on G that screendumps are often easier to understand than text. Dev2 repeats this point for the textual redesign proposals; text can be difficult to understand, and an illustration of the redesign proposals as support for the text is desired.

Dev2 explains how the redesign proposals in R make it easier to understand and accept the critique. He explains how the fact that the evaluator has to illustrate his redesign ideas improves the quality of these. All three developers agree that the justification for the redesign proposal is unnecessary: ‘A good idea should speak for itself’, according to Dev1.

The feature of directly pointing to where the UP occurred received positive comments from all three developers. Dev2 explained how M let him jump from the problem description to an illustration of where the problem occurred, and that this setup was very easy to act on. Dev3 points out a positive feature of G: ‘It gets pinpointed where it [the problem] is’. Formats RGM all include the feature of illustrating where the UP occurred.

4.4 UP characterization

The ratings of different feedback formats may depend on the nature of the problems. To investigate this five researchers rated the 35 UPs according to (a) discoverability; how easily they were discovered and (b) complexity; the perceived complexity of fixing the problem. Discoverability was coded according to the scale *perceivable, actionable* and *constructable* (Cockton & Woolrych, 2001). Complexity was coded with inspiration from (Hornbæk & Frøkjær, 2004) using a three-step scale comprising *complex, medium-sized* and *simple* problems. The average comlexity-discoverability ratio is shown in Table 4, and suggests that the UPs used in this study are mostly simple and perceivable/actionable.

To get an impression of whether the most heavyweight UPs were rated differently than the rest of the UPs we studied the ratings of the six UPs from the bottom-right corner of Table 3. In average the heavyweight UPs were rated 8% lower than the rest of the UPs pre use, though this difference is not significant, $F(1,173)=1.757$, $p>.1$.

4.5 Low answering rate for PFs

In average each developer answered 95% of the questions pre use and 98.5% post use. The only apparent pattern in the unanswered questions was a low answering rate for PFs. This phenomenon can be explained by the fact that three of the five questions specifically concerned UPs. In the interviews the developers expressed general satisfaction with receiving PFs, and pointed to the fact that it is nice to know which parts of the system that work and should not be changed. Dev2 also mentioned the psychological effect of combining negative with positive feedback in order for the critique to be ‘easier to swallow’.

5 Discussion

5.1 Comparing feedback formats

Our explorative study suggest that the multimedia presentation, the GUI binder and the redesign proposals were generally seen as useful input to developers’ work, whereas the human-centered stories were not well received. The problem list was generally rated lower than the three top formats and higher than the human-centered stories.

The explorative study suggests that feedback serve several functions, which change over time. Understanding the problem and being convinced about it is of initial importance to feedback. Information about a problem’s context plays a role for both understanding and a problem’s ability to convince. It elaborates the problem, making it easier to understand, and provides information on what caused the problem thus making it more convincing. When the developer is convinced about the problem and understands it, whether the feedback is easy to use gains importance. Ease of use and thorough contextual information seem quickly to conflict however. When the developer has worked with the problem for a while the feedback finally needs to serve as a reminder to the developer. Below we discuss how the five feedback formats relates to these issues.

The problem list is generally rated lower than the GUI binder, the multimedia presentation and redesign proposals, suggesting that the most commonly used feedback format is not the most effective one. Descriptions of problems seem best suited for communicating simple and uncontroversial UPs where no contextual information is needed. We argue that some of the recommendations to improve problem lists such as ‘be more positive, clear, precise and respectful’ (Dumas, 1989) do not fully address the challenges associated with the problems list. Problem lists do not provide any explanations to bolster its problem description and the format’s ability to convince seems mostly to rest on the evaluator’s ethos and assertiveness. We found that developers used severity ratings to assess the evaluator’s credibility and conclude that well argued severity ratings make problem lists more credible.

The GUI binder, which can be produced at fairly low cost, is generally rated equal to the multimedia presentation and redesign proposals. Seemingly, the context provided by the annotated screendumps is valued greatly by developers either as facilitating a better understanding of the problem. The GUI binder only shows where the problem occurred and gives no information about what led to the problem e.g., compared to the multimedia presentation. This suggests that information about problem occurrence is more important to developers than contextual feedback about for instance users’ interactions with the system.

The multimedia presentation proved less convincing than suggested by the literature on highlights videos. ‘Seeing is believing’ is a common argument for videos (Desurvire & Thomas, 1993) but our explorative study suggest that other formats are equally convincing. Developers call for an easier access to contextual information than video. However, this critique acknowledges contextual information like the one presented by the videos in the multimedia presentation as being important to understanding the problems.

The high ratings of redesign proposals suggest that they serve as a valuable elaboration of the problem description that makes the UP more understandable to developers. This supports the findings of (Hornbæk & Frøkjær, 2005) and suggests that the quality of descriptions of even fairly simple problems is heightened by redesign proposals. The psychological effect of receiving constructive suggestions rather than negative criticism may explain why developers find the format convincing, an important quality overlooked by (Hornbæk & Frøkjær, 2005).

Human-centered stories may perform poorly on the feedback dimension of understanding because they demand the reader to analyze and interpret the narrative before being able to understand the problem. The fictional style of the presentation might also be problematic since it apparently is found unconvincing by developers, something that might be addressed by modifying the narrative style of writing. Human-centered stories are not designed for providing feedback on usability problems however.

5.2 Feedback issues of importance to developers

The explorative study suggests that developers rate UP that they agree with higher than the ones they do not agree with. This finding underlines the importance of feedback formats' ability to convince. Easily fixed problems seem also to be rated higher than problems that are not easily fixed. This finding supports reports on how developers have a tendency to favour the problems easiest to correct (Dumas & Redish, 1999). We were surprised to find that heavyweight UPs were rated lower than lightweight ones since we expected heavyweight problems to be of more importance to system development and thus to developers.

Developers value contextual information, which may explain why the multimedia presentation, GUI binder and redesign proposals, which all describe context such as problem occurrence, are initially preferred by developers. The need for contextual information is linked to developers' wish to investigating certain UPs in depth in order to obtain a better understanding of the problem or to search for convincing factors about the problem.

5.3 Differences in pre and post ratings

The differences in how developers rate feedback formats diminish after they have worked with the feedback items. Since we expected the developers to familiarize themselves with and develop preferences for certain formats during their work with the UPs, we were surprised to find that the post use rating showed no significant differences between the ratings of formats.

Developers assessed the same questions before and after having worked with the problems. We hypothesize that the questions were perceived differently pre and post use. As we have no way of knowing, we will refrain from speculating what the difference in meaning is. We dare to conclude that when first learning about and having to understand a problem the GUI binder, the multimedia presentation and redesign proposals are superior to problems lists and human-centered stories. However, any of the formats serve as a reminder of that problem.

The change in the role and rating of feedback over time suggests that studies solely concerning pre use evaluation results are problematic. We suggest future work on the various stages and roles of feedback.

5.4 Ideas for improving feedback

Developers seem sensitive to information overload and we need to investigate how thorough contextual information can be presented in the least overwhelming manner. A multimedia presentation providing the possibility to study relevant and discard irrelevant information might be a solution. Indexed videos might speed up navigation in a short highlights video, but the solution does not remedy that some problems are poorly explained in a short video. Summaries of the videos modeled over human-centered stories (as a kind of reverse video manuscript) might improve ease of use of contextual information and leave room for the evaluator to elaborate on the problems difficultly illustrated with video. However since human-centered stories are not considered highly convincing this idea is not without problems. Longer problem descriptions with elaboration of the causes of the problem might also improve problem lists. Screendumps of where the problem occurred seem also to be valued information easily produced and may for GUI problems serve as reference for a redesign proposal.

6 Conclusion

The present explorative study aims to investigate how five feedback formats serve to convince and provide an understanding of usability problems. The study suggests that feedback serves multiple purposes, which change over time. Initially feedback needs to convince developers and help them understand the problem. The degree to which a feedback format provides contextual information is crucial to how well it succeeds in convincing and explaining the problem. Having accomplished that, feedback must be easy to use in the developers' daily work. Hereafter it(Dumas & Redish, 1993) mainly serves as a reminder of the usability problem.

Developers rate the multimedia presentation, redesign proposals and the GUI binder with annotated screendumps highest on first hand impression. However, after having worked with the feedback developers rate problem lists, the GUI binder, the multimedia presentation, redesign proposals and the scenario format human-centered stories alike. The findings suggest that all feedback formats may serve as a reminder, but only some provide the information needed to be convincing and helpful in understanding the problem. Problem lists, that are perhaps the most common feedback format, do not provide sufficient information to perform well on first hand impressions, and need to include additional information before providing developers with efficient feedback.

Reference List

Butler, M. B. & Ehrlich, K. (1993). Case Study: Lotus Notes 1-2-3 Release 4. *Published Online*.

Coble, J. M., Karat, J., & Kahn, M. G. (1997). Maintaining a focus on user Requirements Throughout the development of Clinical Workstation Software. *Proceedings of the ACM Conference on Human Factors in Computing, 22-27.March 1997*.

Cockton, G. (2006). Focus, Fit and Fervour: Future Factors Beyond Play with the Interplay. *International Journal of Human-Computer Interaction, 21,* 2, 239-250.

Cockton, G. & Woolrych, A. (2001). Understanding Inspection Methods: Lessons from an Assessment of Heuristic Evaluation. In A.Blandford, J. Vanderdonckt, & P. Gray (Eds.), *People and Computers XV - Interaction without Frontiers. Joint Proceedings of HCI2001 and IHM2001* (pp. 171-191). Springer.

Desurvire, H. W. & Thomas, J. C. (1993). Empirisism versus judgement:Comparing user interface evaluation methods on a new telephone-based interface. *ACM SIGCHI Bulletin, 23,* 4 (October).

Dumas, J. & Redish, J. (1999). *A Practical Guide to Usability Testing*. Intellect.

Dumas, J. (1989). Stimulating Change Through Usability Testing. *SIGCHI Bulletin, July 1989, 21,* 1.

Dumas, J., Molich, R., & Jeffries, R. (2004). Business: Describing usability problems: are we sending the right message? *Interactions, 11,* 4, 24-29.

Dumas, J. & Redish, J. (1993). *A Practical Guide to Usability Testing.* Norwood: Ablex.

Frøkjær, E. & Hornbæk, K. (2005). Cooperative Usability Testing: Complementing Usability Tests with User-Supported Interpretation Sessions. *Extended Abstracts of ACM Conference on Human Factors in Computing Systems (CHI 2005),* 1383-1386.

Hornbæk, K. & Frøkjær, E. (2004). Usability Inspection by Metaphors of Human Thinking Compared to Heuristic Evaluation. *International Journal of Human-Computer Interaction, 17,* 3, 357-374.

Hornbæk, K. & Frøkjær, E. (2005). Comparing Usability Problems and Redesign Proposals as Input to Practical Systems Development. *ACM Conference on Human Factors in Computing Systems.*

Hornbæk, K. & Stage, J. (2006). Special issue on the interplay between usability evaluation and unser interaction design. *International Journal of Human-Computer Interaction, 21,* 5.

Jeffries, R. (1993). Usability Problem Reports: Helping Evaluators Communicate Effectively with Developers. In *Usability Inspection Methods* (Mack, Robert L. ;Nielsen, Jakob; ed., pp. 273-294).

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. M. (1991). User Interface Evaluation in the Real World: A Comparison of Four Techniques. *Proceedings of the CHI Conference on Human Factors in Computing Systems,* 119-124.

John, B. E. & Marks, S. J. (1997). Tracking the effectivity of usability evaluation methods. *Behaviour & Information Technology, 16,* 4/5, 188-202.

Kahn, M. G. & Prail, A. (1993). Formal Usability Inspections. In J.Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods* (pp. 141-171).

Kennedy, S. (1989). Using Video in the BNR Usability Lab. *SIGCHI Bulletin, 21,* 2, 92-95.

Law, E. (2006). Evaluating the Downstream Utility of User Tests and Examining the Developer Effect: A Case Study. *International Journal of Human-Computer Interaction, 21,* 2, 147-172.

Mills, C. B. (1987). Usability Testing in the Real World. *SIGCHI Bulletin, 18,* 67-70.

Nayak, N. P., Mrazek, D., & Smith, D. R. (1995). Analyzing and Communicating Usability Data. *SIGCHI Bulletin, 27,* 1, 22-30.

Nielsen, J. (1993). Heuristic Evaluation. In J.Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods* (pp. 25-62).

Nielsen, J. (1994). UPA93 - Usability Professionals Association Annual Meeting 21-23 July 1993, Redmond, WA, USA. *SIGCHI Bulletin, 26,* 2, 29-32.

Nørgaard, M. & Hornbæk, K. (2006). What do usability evaluators do in practice? An explorative study of think-aloud testing. *Proceedings of the 6th ACM Conference on Designing Interactive Systems*, 209-218.

Redish, J., Bias, R. G., Bailey, R., Molich, R., Dumas, J., & Spool, J. M. (2002). Usability in Practice: Formative Usability Evaluations - Evoluiton and Revolution. *Proceedings of the CHI, April 20-25, Minneapolis, Minnesota*.

Schell, D. (1986). Usability testing of screen design: Beyond standards, principles, and guidelines. *Proceedings of the Human Factors Society 30th Meeting, Santa Monica, CA*, 1212-1215.

Sears, A. (1997). Heuristic Walkthroughs: Finding the Problem Without the Noise. *International Journal of Human-Computer Interaction, 9*, 3, 213-234.

Seffah, A. & Andreevskaia, A. (2003). Empowering Software Engineers in Human-Computered Design. *Proceedings of the 25th International Conference on Software Engineering*.

Strom, G. (2003). Perception of Human-centered Stories and Technical Descriptions when Analyzing and Negotiating Requirements. *Proceedings of Human-Computer Interaction, Interact '03*.

Tables and figures (in order of reference)

Figure 1: The study consists of eight steps. The figure show how the usability test (step 1) is followed by analysis of 75 UPs (step 2) and merging these into 40 UPs (step 3). Finally five feedback items are constructed for each UP, a total of 200 items (step 4).

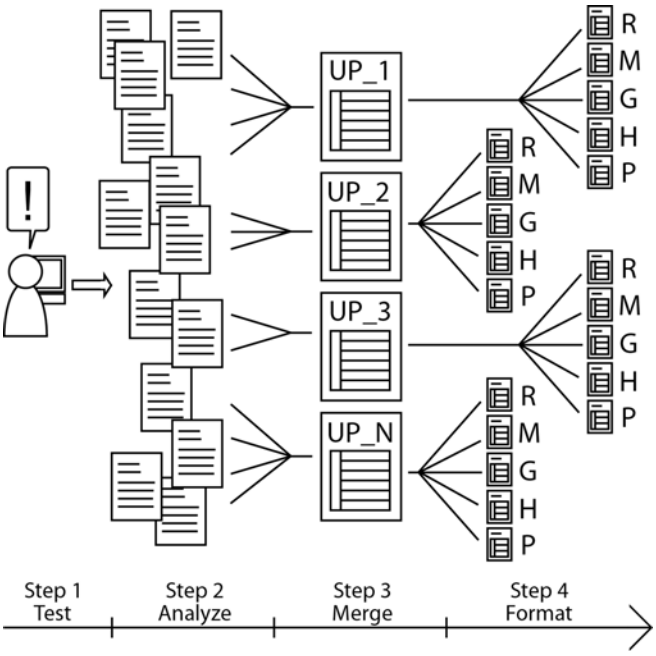

Figure 2: Three developers rate the usefulness of the 200 feedback items (step 5). They then work with selected 40 items (step 6) and re-rate them after completing their work (step 7). The developers are finally interviewed about the use of the formats (step 8).

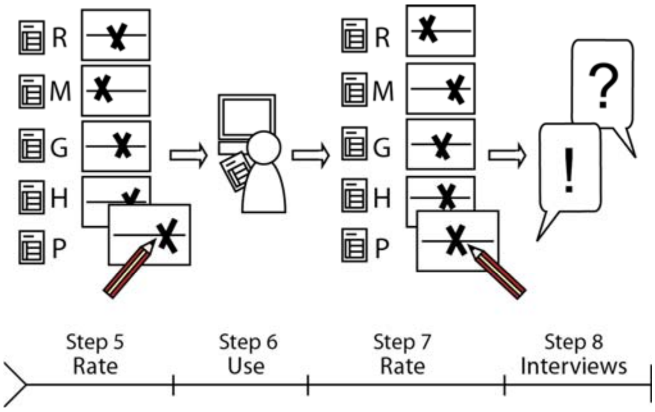

Table 1: Average pre and post use rating of question 1-5 organized by feedback format.

	Problem lists (P)		GUI binder (G)		Multimedia presentation (M)		Redesign proposal (R)		Human-centered stories (H)		F-test	Tukey HSD post hoc tests
	M	SD	M	SD	M	SD	M	SD	M	SD		
Pre use	45.3	12.5	54.0	10.9	53.8	12.1	57.0	11.2	31.6	9.9	*F*(4,170)=28.76, *p*<.001	H<P<MGR
Post use	40.5	14.0	42.1	17.1	50.2	14.6	43.5	12.6	32.4	14.0	*F*(4,30)=1.35, *p*>.3	HPMGR

Table 2: Average pre use ratings of formats PGMRH according to question Q1-Q5. The phenomenon where a format is listed in two significant groups (e.g. for G in Q3) means that the format is neither significantly differetn from one group or the other.

	Problem List (P)		GUI binder (G)		Multimedia presentation (M)		Redesign proposals (R)		Human-centered stories (H)		F-test	Tukey HSD post hoc tests
	M	SD	M	SD	M	SD	M	SD	M	SD		
Q1: How useful is the feedback item to your work on Jobindex.dk? (not useful/ very useful)	58.1	13.0	61.9	11.1	60.0	9.8	63.9	11.3	38.8	12.6	*F*(4,170)= 26.68, *p*<.001	H<PRMG
Q2: How well does the feedback item help you understand the problem? (poorly/very well)	59.1	16.4	67.6	11.8	69.0	14.7	72.8	10.2	41.3	14.3	*F*(4,170)= 29.57, *p*<.001	H<P<RMG
Q3: How well does the feedback item help you solve the problem? (poorly/very well)	28.5	12.7	45.2	15.7	43.1	17.5	50.8	17.0	18.4	8.8	*F*(4,170)= 28.87, *p*<.001	H<P<MG<GR
Q4: How convinced are you that this is a problem? (poorly/very well)	44.6	15.3	50.6	14.4	54.0	13.8	51.0	13.8	32.7	13.5	*F*(4,170)= 12.46, *p*<.001	H<PRG<RGM
Q5: How easy is the feedback item to use in your work on Jobindex.dk? (difficult/very easy)	36.0	15.1	44.7	14.4	43.0	15.7	46.4	14.7	26.8	8.9	*F*(4,170)= 11.73, *p*<.001	H<P<RMG

Table 3: The feedback formats’ strengths and weaknesses

	Pros	Cons
P	Provides short and sufficient information about simple UPs. Ratings of severity.	Does not describe context of UP. A bit too short to describe problems fully.
G	Points to where the UP should be fixed. Screendumps are concrete and often easier to understand than text.	The problem’s context and triggers need to be elaborated. An illustration of the redesign proposal is lacking.
M	Video is credible and persuasive. Quick and easy to use.	‘Overkill’ to describe simple problems with video. Video is too time consuming and it is difficult to get a quick overview of the video
R	Helps solve the problem well. Illustrations improve quality of redesign proposals.	The problem’s context and triggers are not explained well. A justification is unnecessary.
H	Provides information about the context of a UP. Shows where you ‘loose’ the user in the interaction.	‘Overkill’ – it is not a simple way to present a problem. There is a lot of ‘noise’. Time consuming to read and interpret.

Figure 3: It is clear how developers generally rate the 35 UPs lower after having worked with them. The only exception to this trend is H, which in general receives a slightly improved rating.

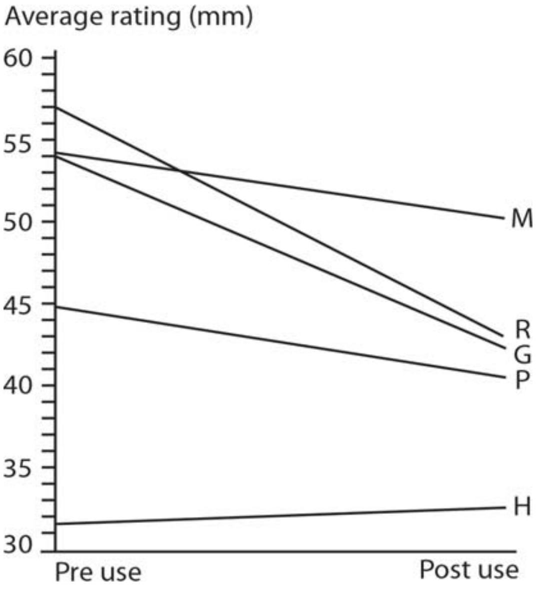

Table 4: The 35 UPs complexity and discoverability clutter in the top left corner suggesting that the UPs used in this study are fairly easy to spot and simple to mend.

	Perceivable	**Actionable**	**Constructable**
Simple	11	15	0
Middle	3	4	1
Complex	0	1	0