IESE

Fraunhofer Institut
Experimentelles
Software Engineering

Useful Design Recommendations from a Pattern-based Usability Inspection: Empirical Evidence

Authors:
Sabine Niebuhr
Martin Schmettow

Submitted for Publication at
NordiCHI 2006, Oslo

IESE-Report No. 060.06/E
Version 1.0
May 2, 2006

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

Recently we introduced a novel method for usability inspection, which is based on usability patterns [16]. The Usability Pattern Inspection (UPI) is hypothesized to have some advantages compared to established usability inspection methods. In particular it is designed to produce detailed and clear design recommendations in addition to mere identification of usability problems. Here we present the results from two empirical studies to show, that the UPI is effective in giving useful design recommendations.

Keywords: Usability Inspection, Usability Pattern, Experiment, Case Study, Usability Pattern Inspection

Table of Contents

1 Introduction

1.1 Overview on Usability Inspection

Usability Inspection is a best practice activity in the development of interactive software systems. It is a group of techniques to identify possible usability defects in a user interface (UI) by expert evaluators expressing their opinion. This clearly distinguishes from participative techniques like the usability test, where real users are confronted with a user interface and a set of tasks and the existence of a usability defect is inferred by observing the test persons' behavior.

Usability inspection methods make use of various sorts of design or quality rules to sensitize the evaluator for usability defects. These rule sets make the main difference between inspection techniques. They range from very detailed descriptions (e.g. design reviews based on UI standards guidelines [22]), transporting a significant amount of usability knowledge, to small sets of heuristics, which rather serve as aid memoirs to the evaluator [12]. Another property of usability inspection, as opposed to usability questionnaires and the so called summative usability testing, is that they serve as an instrument of defect detection instead of measuring certain quality criteria of the software, e.g. effectiveness, efficiency and satisfaction.

Another aspect in that inspection techniques differ is the rigidness of the procedure. Some techniques have a very well defined procedure with quite specific decisions to make, while others more on unrestricted exploration of the evaluator.

The probably most often applied inspection technique is the Heuristic Evaluation (HE), which has a very lax procedure and uses a set of 10 to 12 quite high-level heuristics [12]. These heuristics were originally derived from informed expert opinion, but variants exist, were the heuristics are derived from HCI theories (for an experimental comparison, see [10]). A much stronger procedure and theory-based rules can be found in the Cognitive Walkthrough technique [21]. This technique is supposed to be excellent in finding defects, which hamper the learnability of a user interface. A method exploiting a very rich body of UI design rules is the standards inspection [22].

In this paper we will present first results from a series of experiments and studies on a novel approach to usability inspection: the Usability Pattern Inspection (UPI) [16]. This method utilizes the modern approach of usability patterns to provide a rich body of usability knowledge to the evaluator. Moreover it has a quite well-defined procedure. One outstanding property of the UPI is the strong emphasis on giving explicit design recommendations instead mere identification of possible defects. We hypothesize that this will lead to a high perceived usefulness of the inspection results on side of the developers.

1.2 Usability Inspection in the Software Development Process

Usability Inspections are an established means for quality management of user interfaces. Their characteristics in identifying defects are sufficiently known. Usually they do not require a technical infrastructure like usability testing, which has on the other hand proved to by more effective, and are in general quite inexpensive to conduct..

In certain development situations, especially in project oriented development, it is difficult to have real users participating in the evaluation, for example, when there are only few users and the customer cannot dispense them for usability testing. Then there is no other choice than having an expert taking the perspective of the user.

Another problem with usability testing can be, that sometimes severe defects inhibit the identification of other defects. Expert evaluators are likely more robust against these inhibitors, which often occur in incomplete products. Thus, inspections can be more efficiently applied in early design stages, which is very crucial for software development economics, since it is fact, that late defect elimination costs are an order of a magnitude higher than finding them in early artifacts [2].

These advantages in economy and flexibility make usability inspections the method of choice, especially for small and medium software companies.

1.3 Quality Criteria of Usability Inspection methods

There are several good reasons to investigate measurable quality characteristics of usability evaluation methods than the advantages claimed above. One obvious reason is, that if there is the choice between methods to be implemented in a development process, only rigidly measured quality criteria can help an objective decision. Another reason has to do with risk management: If usability has a high criticality for business goals (e.g. in web shops), then a best effort strategy (also labeled as "discount usability" [14]) does not suffice. Instead, in order to guarantee the certainty of (near) zero usability defects, detailed characteristics of the method at hand have to be known. This should in our opinion be at least: The probability of detecting a defect by one evaluator and the class of usability defects the method effectively detects.

There seem to be (at least) two paradigms of determining the quality of formative usability evaluation methods in HCI method research:

From classical test theory the measures objectivity, reliability, validity and economy are derived. It is to remark here, that the underlying logical structure of a psychometric test is quite different from the case of formative usability evaluation. In classical tests a number of results of tasks are summarized to a single score, which can be used to compare tested subjects. In formative usability evaluation the situation is more like a signal detection experiment, where defects are the signals. Since the quality criteria of classical testing are based on a strict axiomatic system (see [5] or any other introduction to psychological tests), their transfer to usability evaluation is restricted to analogy alone.

Another approach to evaluation of inspection method is derived from biology, where the true size of a population is estimated by the so-called capture-recapture models. Some simpler from this family of models were also applied to usability evaluation [9, 19], and have a tradition in software inspection research [11]. However, the focus in these studies often is to determine, how many evaluators are needed and how many defects are left undetected. But basically these models estimate the probability of one inspector detecting a defect with a certain method. This probability measure can be used to exactly compare different methods and to plan their appliance in practice according to given needs.

Nevertheless, both approaches concentrate on criteria internal to the inspection procedure. The effects on later processes in the development process are often not considered. As these might be very crucial for client goals in professional usability business, we concentrate our evaluation of the UPI on aspects of the most client-side property, which is the usefulness.

1.4 Perceived usefulness

Every artifact that is produced in the development process, but not taken into account in later steps, is simply useless. Although this is a very plausible statement, studies considering the aspect of effectively communicating inspection results to later process stakeholders are quite rare in the HCI scientific literature. In a rough inquiry we found one study, where a systematic study on this topic was conducted [8]. Of course, there exist a number of best practice recommendations in several usability text books (e.g. [15]) and even a Common Industry Format for Usability Test Reports (CIF-UTR) for usability testing [1], but these are bare of empirical evidence.

Because the UPI, which will be outlined in the next chapter was designed with the perspectives of software developers in mind, we were interested in how it performs in this most client-oriented quality aspect.

With perceived usefulness, we mean here the ease of use and effort needed to understand the inspection results and to follow the given advice. Our hypothesis is, that the perceived usefulness of inspection presentation is influenced by at least the following aspects:

- The *structure of the report* influences how easy the report (or presentation) can be read and navigated, where this is likely dependant on the role-specific goals. This is for example reflected in the structure of the CIF-UTR , where an explicit management summary exists.
- Since the defect elimination can be costly, the results have to reflect this by allowing an *economical decision*. This is reflected in the severity estimates to usability defects, which give the defect elimination a certain value. Admittedly, the UPI does not provide an explicit procedure for estimating a defects severity, yet.
- The *language* of the inspection results should be adequate to be easily understood by non-usability experts.
- The *richness* of defect description should be well-balanced. It should not waste time with lengthy texts but must be verbose enough to be understood and accepted by the developers.
- As the inspection results are usually a critic on the UI design, *intimidating communication has be avoided*.
- The report of an identified defect can be appreciated with *added values*. This is in particular to give recommendations for better design alternatives. This does not only save the developers resources, but can also make the defect identification more comprehensible and convincing.

As stated above, it was an explicit design goal of the UPI method to provide added value by giving explicit design recommendations. Accordingly this is the main focus here.

2 Outline of the Usability Pattern Inspection

As the UPI is a method with very well-defined procedures and artifacts, an exhaustive description is clearly beyond the scope of this paper. We will instead describe the basic aspects only.

2.1 Usability Patterns as source of usability knowledge

Usability patterns are on the way to be the format of choice in describing constructive usability knowledge. There already exist several larger collections for general UI domains, like traditional GUIs [18], websites [7] and even UIs on mobile devices [6, 20]. For the appliance during design construction they are especially useful of several reasons, the most important thereof are:

- Patterns have a problem-oriented format, in that they always state a specific design problem and then describe a best-practice solution.
- Patterns are most often collected in so called pattern languages, where classifications and interrelations ease the identification and combination of adequate patterns.
- Patterns are usually on a medium level on abstraction: abstract enough to cover a wide range of design situations, but concrete enough to be easily followed by the developer.
- Patterns are usually written in the language of designers and developers, not human-computer interaction.

In the UPI, however, patterns are in their purpose turned upside down, in that they are used to identify design weaknesses. This follows the simplified argument of "When a UI designer did not use a recommended solution in a specific design situation, than this is likely to result in a usability weakness". This is of course not too obvious, because the fact, that *one* good solution was not chosen, is logically not equivalent with that *no* good solution was chosen. Anyhow, from our experience so far we are optimistic, that patterns are a valid means of identifying usability defects.

The main pragmatic problem of using patterns for an inspection is the mere amount of a pattern collection. For example the pattern collection used in the experiment described later was a staple of about 50 sheets, while the heuristics used in the HE fit on just 2 sheets. This problem was solved with an additional classification of patterns for more efficient selection of patterns that might match. The patterns used for inspection were independently classified by two experts to a set of abstract user activities.

2.2 Inspection procedure

The basic idea behind the UPI was already indicated above; in more specific this is: When during the inspection a specific design situation is identified and none of those patterns have been applied, that claim to be applicable to that situation, than a usability defect is identified.

The detailed checking procedure at a certain step in the Inspection outlines as follows:

1. The inspector identifies the most likely user activity at that step by monitoring her own behavior (This is done self-monitoring).
2. *Example:* To find a record in a table of bibliographic references requires the user to "Search data".
3. She selects all pattern from the collection, which are classified to this user activity.
4. *Example*: The patterns "Search Box", "Sortable Table", "Hierarchical Set" (from [18]) are selected
5. She identifies patterns, which exactly match on the design problem at hand.
6. *Example*: As bibliographic references can well be presented in a table-like data structure, but not hierarchical; "Sortable Table" is taken into further account, while "Hierarchical Set" is discarded.
7. She compares the actual UI design with the pattern and records matches and mismatches.
8. *Example*: The data is indeed presented as described in "Sortable Table". But there is no "Search Box".
9. She derives recommendations for the improvement of the design from the pattern.
10. *Example*: "In the left upper corner of the dialogue should be a search box, which takes arbitrary keywords as input."

The rest of the procedures are similar to other inspection methods in usability and software engineering. Usually the UPI is conducted in several single sessions followed by a reviewer meeting. The path, the evaluator takes in the application, is strictly defined and is usually derived from user manuals or requirements specifications (e.g. use cases).

2.3 “Usefulness features” of the UPI

2.3.1 Recommendations

The UPI method makes use of usability patterns, which were originally meant as construction guidelines. Thus they are an excellent means for adding concrete and well founded design recommendations to the identified defects. There are several variants of giving the recommendations. The least effort variant is to simply deliver the referenced patterns with the report and let the developer derive a concrete design enhancement on his own. This can be problematic depending on the comprehensibility of the used patterns. While some pattern collections give very short and clear statements and illustrate them with nice examples and pictures (e.g. [18]), others describe them prominently textual (e.g. [17]). Nevertheless, with highly-skilled developers this might be feasible.

The better alternative is to give a short recommendation together with the defect. Thereby the evaluator should not restrain to just rephrasing the pattern, but draw the patterns message to the design problem at hand.

2.3.2 Positive feedback

As we argued above, it is important to communicate the inspection results in a non intimidating way. It is further a widely accepted practice of “good criticizing” to also state the positive aspects and at best to state them first. This was an explicit design goal of the UPI. The evaluator is held to not only record the mismatches, but also the matches. This is the basis for positive feedback to the developers. In the presentation of inspection results, this can be at least given as a statistic, e.g. the percentage of patterns that were correctly used. If the presentation or report is a detailed walkthrough, the positive aspects for each UI dialogue can be stated before the critics.

2.3.3 Presentation of results

While positive feedback and concrete recommendations are inherent features of the UPI, we also use a set of guidelines for the presentation of the results, which can in principal be applied with other evaluation methods as well. We call these guidelines the “golden rules to usability results presentation”. These rules are in particular:

1. Be aware in advance to the presentation, that you will criticize the intellectual products of others (the developers).
2. Start your presentation with positive results from the evaluation.

3. Start presenting those defects, that are
   - easy to resolve
   - obvious, or at best already known to the developers. Be especially cautious not to state general critics first and then detailed.
4. Appreciate other goals of the developers, which might be contrary to your recommendation (e.g. technical feasibility, performance or simply effort)
5. Avoid absolute statements (should, must, must not).
6. Give scalable recommendations: A perfect solution can be very expensive, whereas an acceptable solution might suffice.

3 Evaluation Study

The main focus of our empirical evaluation is the added value of the UPI by giving design recommendations. This can be divided in the two aspects:

1. Does the UPI really produce more recommendations than a comparable method?
2. Are the recommendations good in quality, i.e. do they facilitate the elimination of usability defects?

The aspect of productivity is mostly quantitative, whereas the quality aspect can not so easily be counted and is strongly influenced by client's opinion. Accordingly, we decided to put forward two studies: an experimental study to measure the productivity compared to another usability inspection method and a case study to investigate the quality aspects

3.1 Hypotheses

Measuring the productivity of the UPI was done by an experimental comparison of the total number of usability problems found by an experimental group performing the UPI with another experimental group performing the HE. A second dependant variable was the number of recommendations, both experimental groups formulated during the inspection.

Since the usefulness is more related to the given recommendations, only the second aspect is described, and the hypothesis regarding this aspect is proposed as:

H1: An evaluator using the UPI would give more recommendations than an evaluator using the HE.

A further evaluation goal was to get to know, whether the recommendations of the UPI are easy to understand by non-evaluators and non-experts in usability issues, and therefore the following hypothesis was stated:

H2: The recommendations of the UPI are easy to understand.

For proving this aspect more detailed, a further hypothesis states:

H3: The recommendations of the UPI are productive for a rework of the inspected user interface.

While the first hypothesis has been proved by an experiment, the second and third hypotheses have been proved by a case study with a small company. In run-up to the experiment we performed a pilot experiment for time- and complexity estimations.

3.2 Experimental design

The experiment was designed to find out, whether an evaluator using the UPI gives more recommendations than an evaluator using the HE.

3.2.1 Procedure

The experiment started with a short introduction into Usability and into the usability inspection method, the participants would use in the following user interface evaluation of a bibliography management tool. During this inspection they filled in a report, in which they formulated the found usability problems and recommendations to them. After one hour the inspection was ended, and the participants had to fill in a questionnaire about general aspects, such as experiences, as well as specific aspects concerning the used inspection method.

3.2.2 Sampling

Volunteer students and scientific assistants from the computer science department of the University of Kaiserslautern in Germany participated the experiment. They have been divided into two groups by the preferred date the persons had time to take part at the experiment. The first group with 4 test persons was introduced into the UPI, their members were asked to perform; the second experimental group consisted of 6 test persons and was asked to perform the HE.

3.2.3 Independent variables

Since the introduction (excluding the respective inspection method), and the tasks, the test persons had to perform during the inspection as well as the evaluated application and the following questionnaire had been the same in both groups, the only varying aspect was the inspection method the group members had to use during the inspection.

3.2.4 Used material

Material used in the experiment includes the presentation slides for the introduction, the bibliography management tool, which user interface had been evaluated, a task-based walkthrough and a reduced pattern collection and accordingly Heuristics for the inspection, and the questionnaire. The slides for the introduction have been the same for both groups, excepted those describing the usability inspection method and its used usability knowledge. The walkthrough has also been the same for both experimental groups. It includes tasks according to three main use cases of a bibliography management tool, which can be described as entering a new reference, searching for references, and exporting references. The pattern collection was reduced after the pilot experiment up to 49 patterns, which have been taken out of the collections of Tidwell [18] and van Welie [20]. The Heuristics used for the HE in the second experimental group have been taken from the web site of Jakob Nielsen [13].

3.2.5 Measures

The questionnaire's part concerning the used inspection method is leaned on the Technical Acceptance Model (TAM). Its questions are mapped for this situation from questions for perceived usefulness and perceived ease of use from [3]. Since all participants had been Germans, the material – except the patterns – has been translated.

To prove the first hypothesis with this experimental design, we compared the number of recommendations made by the experimental group performing the UPI to the number of recommendations made by the members of the group, which had performed the HE. More formally this can be described as:

#recommendations(UPI) > #recommendations(HE).

3.3 Design of the case study

The case study was designed to find out, whether recommendations are easy to understand and adaptive for a rework of a user interface.

3.3.1 Situation

Since we wanted to know, whether the results of the UPI would be helpful for a redesign of a user interface, we started a case study with a small company that develops software systems for mobile facility management. This company was at that time engaged in a major rework of one of their products. The usability inspection with the UPI was independently conducted parallel to this internal rework.

3.3.2 Procedure

Two evaluators inspected the application's user interface with the UPI. Instead of two separate single evaluation sessions, both inspectors performed the evaluation together. Resulting recommendations from this inspection have been presented to the product manager of the company, who knows the customers' problems as well as the internal redesigns results. For this presentation the so "golden rules" introduced above had been considered.

After the presentation the contact has been interviewed to his perceived usefulness of the results. A third person, who has not involved in the inspection, performed this interview. This was designed to measure whether recommendations can be given with the help of the UPI which can be directly used for a rework of the inspected user interface and whether these recommendations are comprehensible. A further goal was to find out, whether the UPI could add significant input to the rework. Thus, it was also asked, if the reported defects had been new or already known to product manager.

3.3.3 Used material

The case study's material includes the mobile facility application, that should be evaluated, a pattern collection compiled for this inspection, and semi-structured interview guideline for measuring the variables of interest. The mobile application had been made available by the company – readily installed on a pocket pc. The patterns used in the inspection again include the two collections stated above, as well as domain specific patterns for mobile applications from Gibbert [6]. Questions of the half structured interview are again oriented to the Technology Acceptance Model to the perceived usefulness. The questions range from general questions, whether the asked person understood the occurred problems and recommendations that were presented to him, whether he could expect to redesign his product up from these recommendations, to questions concerning several special problems. These problems were taken out from the results presentation held earlier and were structured similarly. For every of these problems the following questions were asked (translated from German):

Do you also see the described problem as a problem?

If yes:

- Was this problem familiar to you?
- Do you agree with the recommendations?

- Is the recommendation concrete enough to realize it?
- Can you imagine to repair this problem in a next version?
- If not, why?
- (technically unfeasible, the realization would be too expensive, it is not a heavy weighted problem)

If not, why?

3.3.4 Measures

To prove the second and the third hypothesis whether recommendations are easy to understand and productive for a rework of a user interface, the member of the company was asked as responsible person for the rework of the inspected application's user interface. The question posed to the first aspect (H2) was one of the detailed questions to the proposed usability problems, whether the contact agrees to the made recommendation. If he did not agree, he was asked (with other questions) whether he did not understand the recommendation. If he did not, it is rated as false, if he did understand it, as true. The number of true hits were related to all proposed recommendations (#recommendations (understood) ~ #recommendations). If there are more than 80% recommendations understood by the non-expert in usability issues, the hypothesis is retained.

$$\frac{\#\, recommendations(understood)}{\#\, recommendations} > 0.8$$

The 80% are an estimation from which point on the recommendations can be described as easy to understand. For the third hypothesis the question was asked, whether the member of the small and medium sized business agrees with the proposal, if it is concrete enough to realize it, and if he thinks this would be realized in the following version. The test statistic was noted with true, if the member of the company said that the recommendation is concrete enough to realize it, and if it is technical realizable. This means, the second question has to be answered with yes and if the third question for the realization in the next version has been answered with no, the reason why has not been the technical one. The realized recommendations we related to the total number of recommendations, like we did it for the understandability of them in the formula above.

3.4 Results

Statistics we could collect in the experiment and the case study have been the reports of the participants, which describe their found problems and recommendations of the user interface inspection and their questionnaires. Out of the case study we have the inspection's report and the answers to the interview. We collected several other measures related to reliability and validity of the UPI, but we focus the result description to strictly those statistics concerning the usefulness of the UPI.

	Number of recommendations		
Group	**Sample size**	**Average**	**Variance**
UPI	4	14.25	23.69
HE	6	6	6
t-test	H_0: $avg_{HE} = avg_{UPI}$ df=8 $t_{□=.05}$=1.86 t=3.55 **t> $t_{□=.05}$: $H_{0\ \text{rejected}}$**		

Table 1: Results of the t-test confirming that the UPI produces more recommendations than the HE

The usefulness concerning the effectiveness of the UPI can be measured with the number of recommendations, an evaluator makes with an inspection method. As hypothesis was proposed that a non-expert in Usability Issues would give more recommendations than a non-expert using the HE (H1). Performing the experiment like it is described above, we counted the number of recommendations given by both groups. Therefore these formulations have not been normalized or rated anyway. The average value of the given recommendations in the group performing the UPI had been avg_{UPI}=14.25 with a variance of var_{UPI} = 23.6875. The average number of recommendations of the group performing the HE had been avg_{HE} = 6 with a variance of var_{HE} = 6. We then calculated a t-test to confirm our hypothesis, that by UPI more recommendations had been given than using the HE. As Table1 shows, we could confirm our hypothesis with an α-error of less than .05. The average number of recommendations given with the UPI is significant larger than with the HE.

In the second hypothesis we stated, that recommendations of the UPI are easy to understand, and proved it with the described case study by counting the understood patterns. Setting them in relation to all presented recommendations, we stated that more than 80% would be understood. 21 recommendations have been presented to the contact, who agreed with and understood 17 of them. Calculating this with the formula described above you can see that 80.095% have been understood. Since 80.1% are more than 80%, our second hypothesis can be retained, that the recommendations given by the UPI are easy to understand.

For the third and last hypothesis we asked the contact in the case study whether the recommendations would be realized. As a result, 17 out of 21 recommendations are concrete enough to be realized. Nevertheless, 7 problems would not be realized, because they were not heavy weighted enough or not technical realizable. For the test statistic this means, that just 14 recommendations can be rated with true and related to the total number of 21 recommendations. The outcome of this calculation is 66.67%, what means that just 2/3 of the recommendations were useful for the company's rework of the user interface. Just considering the 17 recommendations, which are concrete enough to be realized, this usefulness would be calculated with 80.09%. Considering the fact, that some recommendations would not be realized in the next version, the usefulness is calculated with only 66.67%.

A further goal was to find out, whether usability problems have been found by the UPI, which have not been found by the company's redesign and vice versa.

As can be seen in only 3 of 21 the usability problems were previously completely known to the member of the company, 10 of them were partly known, and 6 problems were unknown to him. Only two usability problems were not accepted as problems by the contact.

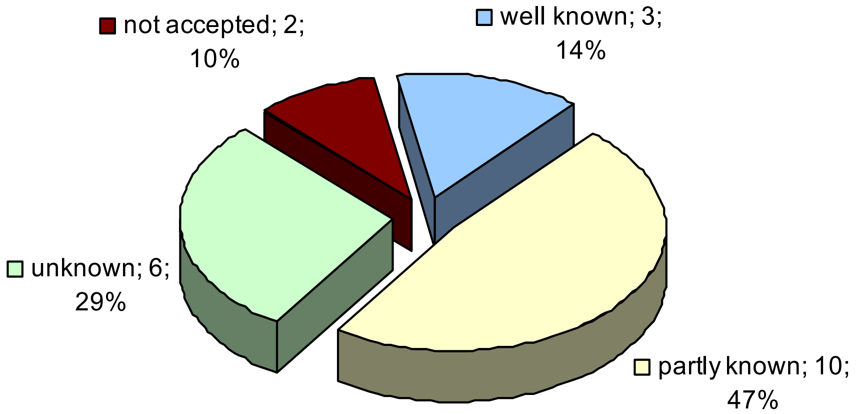

Figure 1: Overview to the high profile of the problems in number of problems and share in the total number of 21 problems

4 Discussion

In this section we will evaluate our hypotheses concerning the usefulness of the UPI in the light of our empirical results. Furthermore we gained some unsystematic experience, which will lead to new research topics. But since in empirical studies there is always the possibility of biased findings and alternative causal relationships, we will first discuss the validity of our studies.

4.1 Threats to validity

4.1.1 Internal validity

Internal validity is given, when the measured effects doubtlessly are caused by the treatment and are not influenced by other factors. This means for the conducted experiment that different results are only caused by the independent variable, which is the performed inspection method. There are a few important threats to internal validity.

The analysis of the third hypothesis can be underestimated since the contact person is influenced by the recommendations given by the application's redesign in parallel. This situation was chosen to have some sensitiveness of the product manager. But in a normal situation more recommendations might be perceived as useful.

Regarding this hypothesis again, it also should be said, that the only the most affected screens with the heaviest weighted faults have been referred to in the interview. As one could expect the less severe defects to be also less convincing, this likely causes an overestimation of the overall accepted recommendations.

4.1.2 External validity

External validity is sufficient, if the result of a sample analysis can be generalized on other persons, situations, or points of time. For the performed experiment this means, that its findings can be generalized from this laboratory situation to industrial or practical situations. Possible threats to external validity are as follows.

An experiment specific aspect is the selection of patterns. For the experiment we had to reduce the amount of patterns available to the evaluators. It is unclear, if this causes an under- or overestimation of results: On the one hand not all patterns were available to identify defects (underestimation), on the other hand the process of pattern selection was made easier than in a real situation. Nevertheless we made sure, that the ratio of matching and non-matching patterns remained the same as in the complete pattern collection.

Regarding the third hypothesis, the selection of problems (based on the most affected screens) presented to the client can lead to an overestimation of the UPI. The omitted usability problems are likely less severe. Since the presentation focused on the part with the heaviest weighted faults and these would be rather eliminated than less heavy weighted problems, a generalization would be a too optimistic calculation and therefore lead to an overestimation of the UPI's results.

4.2 Summary of results

Hypothesis 1 regarding the recommendations given with the UPI has been retained, since the recommendations of the evaluators using the UPI are more than the recommendations given by the members of the HE group. More recommendations enable an easier elimination of problems, if the recommendations are clear enough.

The second hypothesis concerns the understandability of the recommendations made by the UPI. As a result of the case study can be said, that the recommendations of the UPI are easy to understand although the assessing person has been no expert in usability issues. In a further study the generalizability of this aspect should be evaluated.

The last hypothesis concerned the applicability of the recommendations made by the UPI. It can be said, that 2/3 of all considered recommendations could have been used for a redesign of a user interface.

Not directly concerning the hypotheses is the additional value, the UPI has, since at least 29% and at most 76% of the found problems have not been discovered by methods, the company used for its internal redesign. Other problems, which had not been discovered by the UPI, but found by those methods the company had used in parallel, have been mostly those, which only would be found with an "expensive" usability evaluation with real users or with specialists with domain knowledge. Therefore they have been identified as problems which would probably never be found by a discount usability engineering method without considering domain knowledge.

Summarizing the usefulness, the contact of the company senses this usability inspection method as very helpful to find general usability problems in a user interface of an application. He especially appreciates the possibility of adapting and specializing the patterns to the specific company situation and domain. He also thinks, that it would be very cost efficient for the company to let someone perform a usability inspection with this method, who is no usability expert and therefore not as expensive as such a person. Nevertheless he would not rely on this method in all cases: for a final version of a user interface, a further usability test with real users and usability experts would be indispensable. However, this final test would be less expensive, because the general usability problems would have been identified with the UPI in advance.

4.3 Efficacy of UPI in giving recommendations

The results from the case study have shown that the recommendations given are predominantly correct and new to the customer. Furthermore the experiment has proven that the UPI truly outperforms the HE method in terms of recommendations. In fact, one could argue, that this finding is an experimental artifact, because the test persons in the UPI condition were instructed, how to derive recommendations from the pattern, whereas the HE test persons were just asked to give recommendations. But we believe that this is a true feature of the UPI stemming from the constructive nature of patterns.

Additionally the conditions in the experiment were quite controlled, so that the effort in both groups was definitely equal. The fact, that there were many more recommendations given in the same time makes it likely, that there was also an at least sufficient effectiveness in identifying defects with the UPI.

The experimental situation was not too artificial and also the participants have characteristics similar to the intended target group [16]. Thus we are optimistic, that the UPI will perform efficiently in real software development processes, in that it informs UI developers about possible usability problems and alternatives to better design.

4.4 Other findings

Performing the experiment and the case study we made experiences and got starting points to increase the usefulness of the UPI. This appears in variances of the process, combination of tasks in the software development process, and presenting recommendations.

The inspection of the mobile application in context of the case study has been done together by two evaluators. During the inspection we made the experience, that it might be a good combination, that one inspector had experiences with the patterns and the UPI and the other inspector had some experiences with the application. These two experience levels made the inspection very effective: One inspector following an explorative process, focusing on functionality, and the other inspector following a more checklist based process focusing on the appliance of patterns.

Therefore it would be interesting to analyze, whether both inspections – for functionality and for usability – could be combined in an effective and cost efficient way, especially, since the UPI is assumed to be easy to use also for software developers. Maybe this inspection method could be integrated and used in an early process step by developers in a pair inspection in a cost efficient way. In other words: this kind of pair inspection could be an anchor for integrating the method into existing quality management activities such as functionality inspection.

Another finding of the case study was an extension for the "golden rules", to find out the customers prioritization and to center those problems in a presentation, which keeps in mind these priorities. This 7th rule could be formulated as follows:

- If the customer priorities (e.g. the main function areas) are known, consider them in order and level of detail of the presentation – which means that high prioritized issues should be presented more detailed and at first.

This rule obviously conflicts with rule 3 (obvious and easy to resolve defects first). But either there has to be found a

balance between the two or rule 3 should dominate when presenting to the developers while rule 7 applies better when talking to managers.

Another promising finding in the case study was, that the UPI seems to work good with innovative technologies, in this case mobile applications.

Another promising aspect that should be mentioned is the range of applicability of the UPI. Especially it was not clear in advance if the method would work in a highly innovative domain like mobile applications. Further studies will show, if the UPI is truly applicable in every domain, for which pattern exist.

5 Future Prospects

The UPI is an emerging method for usability defect identification. In this paper we have provided empirical evidence for the methods efficiency in that we concentrated on the most "customer-oriented" property that is the perceived usefulness.

We have chosen a mixture of experimental design and case study, which in general should grant a good balance between internal and external validity. Especially external validity is usually restricted in controlled experiments, but at the same time crucial, because we are dealing here with "economic" research questions beyond just testing theories. The technique employed in the case study is in our view well suited to efficiently gain robust data. This might even be done with little overhead in industrial contract studies for further evaluation and improvement of the UPI. A series of case studies with equal measures and some classification variables could results in a quasi-experiment with higher internal validity than a single case study.

In future studies the detailed characteristics of the method will be investigated as this was done in the past for today's established usability evaluation methods like the HE and usability testing. The next step will be to present some first results on the general validity of the defect identification from the same experiment. If this succeeds, a more detailed determination should be done, of what types of defects the UPI is excellent for and for what defects it should be complemented with another method. This will require an external criterion with high validity and broad sensitivity like usability testing or preferable a combination of established methods.

But, as was claimed in [16], the method should be especially useful for industrial practitioners without much usability experience. This has at least two aspects to be considered: First it has to be examined, how sensitive the performance of the method is to the evaluator's experience. This has always been a weakness of the HE with non-experts performing much poorer [4]. We propose, that this can be examined in detail with advanced capture-recapture models [11]. If the UPI, as hypothesized, has a low sensitivity to evaluator experience, a model containing the evaluator capability as a variable should not explain the data better than a simpler model without this factor.

The second aspect regards the ease of implementing the method into existing development processes. The sub-aspect of usefulness of results was already investigated here, but it is at the moment not clear, if developers will easily learn and agree to conduct the UPI and what other context factors have to be considered for successful appliance of the UPI.

Literatur

ANSI, A.N.S.I.-. Common Industry Format for Usability Test Reports, American National Standards Institute, 2001.

Boehm, B.W. and Basili, V.R. Software Defect Reduction Top 10 list. IEEE Computer, 34 (1). 135-137.

Davis, D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Quarterly, 13. 319-340.

Desurvire, H.W. Faster, Cheaper!! Are Usability Inspection Methods as Effective as Empirical Testings? in Nielsen, J. and Mack, R.L. eds. Usability Inspection Methods, John Wiley & Son, 1994, 173-202.

Fischer, G. Einführung in die Theorie psychologischer Tests. Verlag Hans Huber, Bern, 1974.

Gibbert, R. Eine Design Pattern-Sprache für mobile Applikationen mit dem Schwerpunkt Navigationssysteme, 2003.

Graham, I. A Pattern Language for Web Usability, London, 2003.

Jeffries, R. Usability Problem Reports: Helping Evaluators Communicate Effectively with Developers. in Nielsen, J. and Mack, R.L. eds. Usability Inspection Methods, John Wiley & Son, 1994, 273-294.

Law, E.L.-C. and Hvannberg, E.T., Analysis of Combinatorial Effect in International Usability Tests. in CHI 2004, (Vienna, Austria, 2004), 9-16.

Law, E.L.-C. and Hvannberg, E.T., Analysis of Strategies for Improving and Estimating the Effectiveness of Heuristic Evaluation. in NordiCHI '04, (Tampere, Finland, 2004), 241-250.

Miller, J. Estimating the number of remaining defects after inspection, International Software Engineering Research Network, 1998.

Nielsen, J. Heuristic Evaluation. in Mack, N. ed. Usability Inspection Methods, 1994, 25-61.

Nielsen, J. Ten Usability Heuristics.

Nielsen, J. Usability Engineering. Morgan Kaufmann, San Diego, 1993.

Rubin, J. Handbook of Usability Testing. How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons, 1994.

Schmettow, M., Towards a Pattern Based Usability Inspection Method for Industrial Practitioners. in Interact, (2005).

Tidwell, J. COMMON GROUND: A Pattern Language for Human-Computer Interface Design, 1999 (last updated).

Tidwell, J. Designing Interfaces. O'Reilly, 2005.

Virzi, R.A. Refining the Test Phase of Usability Evaluation: How many Subjects is enough? Human Factors, 34 (4). 457-468.

Document Information

Title:	Useful Design Recommendations from a Pattern-based Usability Inspection: Empirical Evidence
Date:	May 2, 2006
Report:	IESE-060.06/E
Status:	Final
Distribution:	Public