A Pattern-based Usability Inspection Method: First Empirical Performance Measures and Future Issues

Martin Schmettow
University of Passau
Information Systems II
Innstr. 43
94032 Passau, Germany
schmettow@web.de

Sabine Niebuhr
University of Kaiserslautern
Software Engineering Research Group (AG SE)
Gottfried-Daimler-Straße
67663 Kaiserslautern, Germany
sabine.niebuhr@iese.fhg.de

ABSTRACT

The Usability Pattern Inspection (UPI) is a new usability inspection method designed for the added downstream utility of producing concrete design recommendations. This paper provides first empirical evidence that UPI measures up to the established inspection method Heuristic Evaluation (HE) regarding defect identification. It is shown that there is also some potential for synergy between UPI and HE. The further research plan of measuring UPI is presented.

Categories and Subject Descriptors

H.5.2 [*User Interfaces (e.g. HCI)*]: Evaluation/methodology

General Terms

Measurement, Human Factors

Keywords

Usability Inspection, Experiment, Evaluation, Usability Patterns

1. INTRODUCTION

In the development of usable interactive software applications, inspection methods play a major role for the early identification of usability defects. Whereas the evaluation of inspection methods has always been of interest in usability research, two major topics have been raised in the last years: Improved approaches for valid measurement of method performance have been triggered by harsh criticism on previous evaluation studies [9] (also, see section 1.2). In response to the "Five users is (not) enough" debate, advanced models for predicting and monitoring evaluation processes have been explored (e.g., [7, 8]). In contrast, there have been only few efforts to enhance methods or design new ones in order to resolve principal issues with existing methods.

One such issue is, that common inspection methods lack support for redesign. This was uncovered by a longitudinal study investigating the value of usability defect reports for the actual fixing of these defects. The alarming result was, that only few previously reported defects were adequately fixed [5]. This might be overcome by integrating design recommendations into inspection methods. A more recent study has shown that reporting design recommendations even has the potential of replacing traditional defect reports [11], as they are at least comparably informative and convincing to "downstream" software developers.

The Usability Pattern Inspection (UPI) method was explicitly designed for reporting concise design recommendations together with the defects, facilitating effective problem fixing in the user interface [15].

1.1 Pattern-based usability inspection

Usually, an inspection method supports experts with a particular set of guidelines for identifying possible defects. For example, the Heuristic Evaluation (HE) – the most common inspection method – supports defect identification with a set of 10-12 heuristics, which have once been acquired from usability experts [14]. One major property of the Usability Pattern Inspection is to provide the inspector with a very rich set of guidelines (usability patterns) on the level of the interaction design. These patterns were collected from several widely known collections, e.g., [16, 18]. It was assumed that the following properties make patterns valuable for usability inspection: First, they are problem-oriented in that they describe the context and specific forces of recurring usability design problems. Second, they are on an intermediate range of abstraction which makes them applicable for a wide range of applications and platforms. And third, they describe established design solutions in detail and confirm them with rationales. This enables the inspector to rationalize on the design level – as opposed to the task or user level. As argued earlier [15], there is a need for evaluation methods, which can efficiently be applied by non-experts in small and medium enterprises (as opposed to highly specialized usability experts in large companies). We believe a design-oriented approach to be more natural for common software practitioners than more abstract guidelines, like heuristics. The focus on design is also the main source of UPI's downstream utility, in that the inspection results contain concrete design recommendations beyond mere defect identification.

Whereas the guidelines employed in UPI are focused on design solutions, the walk-through procedure provides a strong notion of user goals and interaction. First, the inspection process is guided by a predefined set of critical user tasks, which can, for example, be derived from requirements documents. Another sub-procedure of UPI was explicitly designed to let software experts take the perspective of the common user on the level of singular interaction events: The inspector is held to permanently monitor his/her own activities according to a set of 16 predefined abstract user activities. This self-monitoring is further encouraged: It facilitates preselection of applicable pat-

Published by the British Computer Society
Volume 2 Proceedings of the 21st BCS HCI Group Conference HCI 2007,
Lancaster University, 3-7 September 2007
Devina Ramduny-Ellis Dorothy Rachovides (Editors)

terns from the pattern repository via a search interface, because the patterns have been classified to those user activities.

In brief, the UPI inspection procedure is as follows:

1. The inspector selects a user task and starts doing it.
   Example: The search facility of a mobile phone's address book is inspected.
2. Whenever the inspector monitors a change in his current user activity, it is time to inspect.
   Example: After initial orientation the inspector searches for a specific entry.
3. A preselection of matching patterns is received from the pattern collection.
   Example: The inspector selects all patterns about searching.
4. Patterns that match the current situation based on dialog configuration and user interaction are compared with the current design.
   Example: The inspector considers patterns "List Browser" and "Continuous Filter" to match the situation (both from [18]).
5. If the current design deviates from the pattern's proposed solutions, this is denoted as a possible usability defect and ...
   Example: The inspector notes that "List Browser" is implemented, but no feature to filter address entries is present.
6. ... a recommendation for improvement is given according to the pattern.
   Example: The inspector notes that adding a "Continuous Filter" allowed for efficient search on small screens.

1.2 Assessing inspection methods

The primary goal of assessing a usability inspection method is to prove its performance in validly identifying those misconceptions in the UI design that are liable for usage problems. After the criticism above the poor validness of effectiveness studies before 1999 [9], several researchers have established a canonical model of measuring the effectiveness of usability evaluation methods [6, 10]. The basic measures of an evaluation experiment resemble the categories known from the signal detection theory: *Hit* - a true defect was detected, *false alarm* - a defect is denoted with no usage problems truly arising, *correct rejection* - an element is correctly identified as defect-free, and *miss* - a true defect was not identified.

The basic performance criteria are derived from these measures as:

$$\begin{aligned} \text{Thoroughness} &= \frac{\text{number of hits}}{\text{number of real defects}} \\ \text{Validity} &= \frac{\text{number of hits}}{\text{number of hits} + \text{number of false alarms}} \\ \text{Effectiveness} &= \text{Thoroughness} \times \text{Validity} \end{aligned}$$

For exact signal detection measures, complete knowledge of the true defects – those that are liable for observable usage problems – is required. In case of incomplete knowledge, thoroughness is prone to overestimation and validity to underestimation. As complete knowledge of usability defects is very expensive to acquire, two variants of usability testing can be employed in order to estimate the performance measures efficiently: With *falsification usability testing,* the defects identified in an inspection experiment are challenged with a very focused test design (focused tasks, predefined observation set) in order to identify false hits [2]. *Asymptotic usability testing* is employed to estimate the total number of existing defects without the need to identify them all. Whereas the identification of false alarms is required for the computation of validity, it is not necessary to know the complete number of defects if the goal is a comparison of two methods. Thus, the experiment presented here is complemented by a falsification usability test only. Of course, previously undetected defects were also recorded to get a rough idea of the absolute measures for thoroughness.

1.3 Hypotheses

Whereas the performance of HE has been assessed in numerous studies and is known to be fair, UPI is a newly developed method with unknown performance. While this method was designed with added downstream utility in mind, it still has to be shown that it is comparable to HE in the amount of true defects detected:

HYPOTHESIS 1. *The thoroughness with UPI is at least as good as with HE.*

UPI's main feature for identifying defects is to evaluate an existing design against established design solutions. It can thus be argued that a considerable amount of false alarms can be expected with UPI as there might be a design that is working well, although it was not yet captured as a usability pattern. However, the complete procedure of evaluating a solution supports a strong user perspective and additional decision points to mitigate this problem. Additionally, HE is also known to produce a considerable amount of false alarms [1], thus:

HYPOTHESIS 2. *The validity of UPI is at least as good as with HE.*

The UPI method provides guidelines quite different from HE and thus might have qualitatively different capabilities and limitations to capture defects and produce false alarms. From the perspective of signal detection, this method-specific detection profile cannot be efficiently broadened with larger inspection groups. In contrast, it is necessary to mix methods with different profiles in order to average out the method-specific bias. Thus, if there is a method-specific detection profile:

HYPOTHESIS 3. *Mixed method inspection groups perform better than pure method groups.*

2. METHOD

2.1 The inspection experiment

To gather the inspection performance data, we asked 10 persons (male students of computer science and researcher holding a diploma in computer science, ages between 25 and 34) to evaluate a bibliography management tool. Randomly divided into two groups they were given three typical user tasks for evaluation (adding a reference, searching for a reference and exporting a list of references). Only one participant had previous experience with usability evaluations and was assigned to the HE group which is conservative regarding our hypotheses. The settings for both groups were the same, except the usability inspection method: The first group performed UPI ($n_{\text{UPI}} = 4$), the second group performed HE ($n_{\text{HE}} = 6$) as a control group.[1] At the beginning, we gave both groups a short introduction to the usability inspection method and the bibliography management tool. The main part was the evaluation of the tool with the introduced inspection method which ended after one hour (controlled time). The participants had to report their findings in a structured template, comprising the defect description, the dialog element, the heuristic or pattern applied, and additional comments. Also, a few questionnaires were presented to obtain the participants personal data and impressions of using the method.[2]

[1] The uneven group size was due to some appointment conflict of one of the participants.

[2] These results will not be presented here.

2.2 The falsification usability test

The falsification test study followed the procedure described by [2] with some enhancements towards adaptive testing. First, a small set (2-6) of observable usage problems was compiled for each defect. Then, a set of tasks was prepared that was supposed to challenge each predicted defect at least twice. During the test, a structured observation protocol was used by the observer to record each time a predicted usage problem was observed. As it suffices to verify each defect only once[3], the observation protocol could be adaptively reduced after each session. After the second session, some "exhausted" tasks could be replaced in order to put a stronger "stress" on the remaining defects. Additionally, the observer collected new defects in a separate protocol. Because defect identification in usability tests is an asymptotic process [17], a stop rule had to be defined for adding further sessions. We decided to finish the study when a session gained no more than one new verified defect, which happened with the fourth session. All four participants had an academic background but did not have any previous experience with this particular bibliography management tool.

3. RESULTS

During the inspection experiment, 48 individual defects were proposed by the participants, 22 in the UPI group and 28 in the HE group. Via falsification testing, 35 defects could be validated, whereas 13 were considered as false alarms. Additional 35 defects were identified in the falsification test which have not been detected in either of the inspection groups. Table 1 shows the basic defect detection measures on the group level. Note that counting true rejections is not applicable here, as there is no data available on the negative decisions of the inspectors. The larger HE group identified slightly more defects but also produced more false alarms. Both groups missed a considerable number of defects.

Table 1. Basic defect detection measures for the two inspection groups

	Hits		False Alarms		Misses	
	UPI	HE	UPI	HE	UPI	HE
Group Count	22	28	5	11	50	44
Intersection	15		3		37	

The three performance criteria will be compared using the means of individual inspector performance. As the sample size is quite small, no statistically relevant differences could be found via comparison of means (t-test) or variances (F-test). As was argued above, the thoroughness reported here is a comparative measure, as no asymptotic usability testing was conducted. The set of 70 defects identified in the experiment or the falsification test is the reference for the thoroughness measures reported here. As table 2 shows, individual thoroughness is quite low in both methods, but there is no considerable difference, neither in mean nor in variance (reported as standard deviation). The validity measures were obtained with the set of proposed defects that remained unverified after the falsification testing. Table 2 shows, that both methods perform equally well in avoiding falsely predicted defects. However, it might be the case that there are larger individual differences with HE. This also shows up in the larger number of overally introduced false alarms, which summed up to 11 in the HE group. But this effect is far from being statistically verifiable. Accordingly, both methods perform equally well regarding the derived measure of effectiveness.

The third hypothesis stated that HE and UPI differ in the types of defects they capture, so that mixed method groups perform better than pure method groups. This was analyzed by simulating all possible inspection groups with four inspectors from the whole sample. This results in pure HE ($n_{\text{HE4}} = 15$) and UPI groups ($n_{\text{UPI4}} = 1$) and mixed groups with either one, two or three inspectors from each condition ($n_{\text{Mix4}} = 194$). For each single group, the hits, misses, and false alarms were combined. Table 3 shows the resulting performance criteria of the three conditions in the simulation. As expected from the previous analysis, there is only a marginal difference between the pure groups. But indeed, there appears to be a slight advantage of mixed groups, mostly caused by a gain in thoroughness.

[3] This only holds, when mere identification is of interest, for severity estimation more observations were needed.

Table 2. Comparison of performance measures for the two methods per individual inspectors

	Thoroughness		Validity		Effectiveness	
	UPI	HE	UPI	HE	UPI	HE
Mean	.144	.150	.834	.843	0.121	.128
SD	.058	.052	.039	.084	.0522	.050

Table 3. Performance measures (means) from simulated groups of four inspectors

	Thoroughness	Validity	Effectiveness
UPI	.306	.815	.249
HE	.332	.761	.253
Mixed	.350	.788	.276

4. DISCUSSION

We have conducted a comparative method evaluation study, which adhered to recent standards regarding study design and performance criteria. However, the study is only preliminary because the sample size was much too small to gain any statistically relevant results.

First of all, we achieved quite low values for the thoroughness of both methods, which would not satisfy real applications (about 10 inspectors were needed to capture 80% of the defects with a thoroughness of .15). The reason for these low values probably is, that the experiment was conducted under restricted laboratory conditions: The participants were non-experts, introduced to the method and the tested application in only one hour, and had just another hour for the inspection. Especially UPI has quite a detailed procedure and also a large body of guidelines, which is usually trained in at least half a day. It can be expected that a more thorough training and supervised practice would enhance the performance significantly. Yet, validity was fair in our study.

Regarding the performance comparison, there were no considerable differences between the new UPI method and HE regarding thoroughness, validity and effectiveness on an individual level. To draw a careful conclusion: Our study provides no arguments for practitioners to hesitate using UPI instead of HE and, in turn, benefit from the additional downstream utility of design recommendations.

We also found preliminary support for the advantage of using a deliberate mixture of methods in inspection processes. In our simulation of inspection groups, a method mixture seemed to broaden the perspective in that more different defects were detected. This happened without sacrificing the validity too much, which was recently reported as a problematic side effect of employing larger inspector groups with a single method [1]. This result also supports our hypothesis that UPI and HE have differing detection profiles and serves as one starting point for the further research agenda, which will be outlined below.

5. FURTHER RESEARCH

The experiment reported was our first effort to systematically evaluate UPI. However, the results are far from being sufficient to employ

the method based on precise economical considerations. In order to prove the value of UPI (and any other evaluation method), it has to be assessed under more realistic conditions. This can, to some extent, be accomplished by modifying the laboratory procedure, but for considerable ecological validity, our laboratory studies will be complemented by a few industrial case studies. These will also challenge the assumed downstream value of design recommendations.

One general limitation is due to the foundations for measurement of inspection performance. The measures gained with typical inspection experiments suffice to compare two methods only if they are assessed under completely equal conditions. But they suffer from severe restrictions regarding generalizability. It is a lesson learned from the "Five users is (not) enough" debate, that simple probability measures for the defect detection capability of a method are not a reliable base for predicting the outcome evaluation processes (e.g., [7]). But predictability is a necessary precondition for giving (credible) guarantees to stakeholders who request a usability evaluation. A strikingly simple measure for the predictability of inspection processes is the variance in performance between inspectors [8]. For UPI currently a study with a larger sample size is in progress for comparing it to HE regarding performance variance. The hypothesis is that UPI produces less variance because it provides a better defined procedure and detailed guidelines to the inspectors.

The claim for predictability is also tightly connected to the problem of the reliability of evaluation methods. It was recently argued that high reliability might not be desirable, because it sacrifices the inspector's different viewpoints as a source of thoroughness [10]. Regarding process predictability, this is, in our opinion, a problematic argument, unless the method-independent contribution of each inspector is known (e.g., from individual performance tests), valid, and deliberately assembled. A better alternative is probably to have different reliable but limited-in-scope evaluation methods at hand. This would allow for a purposeful mixture of several methods in the inspection process in order to gain a broad defect identification profile. A similar approach of separating perspectives has already proved successful for perspective-based inspections in Software Engineering [4]. A study is currently planned to further examine what was started here – advantages of inspection processes where UPI is complemented by another evaluation method.

For usability evaluations, it has recently been observed that specific perspectives (or kinds of knowledge) contribute differently to detection performance [3]. It is likely that UPI has a specificity for defect types in that it provides design knowledge in the first place. A study is currently planned to investigate the performance profile of UPI regarding different defect types. A first candidate for defect classification is the Usability Problem Taxonomy [12], but also the recent efforts of the MAUSE initiative [13] will be considered. A defect type related performance profile will allow practitioners to deliberately choose (or avoid) a particular method for certain kinds of expected defects.

These goals of measuring and profiling UPI are quite ambitious. They might eventually foster new measurement approaches and deeper insights regarding the anatomy of usability inspection processes in general.

6. REFERENCES

[1] ALAN WOOLRYCH, AND GILBERT COCKTON. Testing a conjecture based on the DR-AR model of usability inspection method effectiveness. In *Proceedings of the British 16th HCI Conference* (Sept. 2002), Vol. 2.

[2] ALAN WOOLRYCH, GILBERT COCKTON, AND MARK HINDMARCH. Falsification Testing for Usability Inspection Method Assessment. In *Proceedings of the HCI 2004* (2004).

[3] ALAN WOOLRYCH, GILBERT COCKTON, AND MARK HINDMARCH. Knowledge Resources in Usability Inspection. In *Proceedings of the HCI 2005* (2005).

[4] BASILI, V. R., GREEN, S., LAITENBERGER, O., LANUBILE, F., SHULL, F., SORUMGARD, S., AND ZELKOWITZ, M. V. The empirical investigation of perspective-based reading. *Empirical Software Engineering 1*, 2 (1996), 133–164.

[5] BONNIE E. JOHN, AND STEVEN J. MARKS. Tracking the Effectiveness of Usability Evaluation Methods. *Behaviour & Information Technology 16* (1997), 188–202.

[6] COCKTON, G., LAVERY, D., AND WOOLRYCH, A. Inspection-based evaluations. In *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications*, J. A. Jacko and A. Sears, Eds. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 2003, pp. 1118–1138.

[7] DAVID A. CAULTON. Relaxing the homogeneity asssumption in usability testing. *Behaviour & Information Technology 20*, 1 (2001), 1–7.

[8] FAULKNER, L. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments & Computers 35*, 3 (2003), 379–383.

[9] GRAY, W. D., AND SALZMAN, M. C. Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction 13*, 3 (1998), 203–261.

[10] HARTSON, H. R., ANDRE, T. S., AND WILLIGES, R. C. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction 15*, 1 (2003), 145–181.

[11] KASPER HORNBÆK, AND ERIK FRØKJÆR. Comparing usability problems and redesign proposals as input to practical systems development. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2005), ACM Press, pp. 391–400.

[12] KEENAN, S. L., HARTSON, H. R., KAFURA, D. G., AND SCHULMAN, R. S. The usability problem taxonomy: A framework for classification and analysis. *Empirical Softw. Engg. 4*, 1 (1999), 71–104.

[13] LAW, E. L.-C., HVANNBERG, E. T., COCKTON, G., PALANQUE, P. A., SCAPIN, D. L., SPRINGETT, M., STARY, C., AND VANDERDONCKT, J. Towards the maturation of IT usability evaluation (MAUSE). In *INTERACT* (2005), M. F. Costabile and F. Paternò, Eds., vol. 3585 of *Lecture Notes in Computer Science*, Springer, pp. 1134–1137.

[14] NIELSEN, J. Enhancing the explanatory power of usability heuristics. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1994), ACM Press, pp. 152–158.

[15] SCHMETTOW, M. Towards a pattern based usability inspection method for industrial practitioners. In *Proceedings of the Workshop on Integrating Software Engineering and Usability Engineering (held on Interact 2005)* (2005). http://www.se-hci.org/bridging/interact2005/03_Schmettow_Towards_UPI.pdf

[16] TIDWELL, J. *Designing Interfaces*. O'Reilly, 2005.

[17] VIRZI, R. A. Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors 34*, 4 (1992), 457–468.

[18] WELIE, M. V. Patterns in interaction design, 2003. http://www.welie.com/