# Perspective-based Usability Inspection: An Empirical Validation of Efficacy

Zhijun Zhang, Victor Basili, and Ben Shneiderman
Department of Computer Science
University of Maryland
College Park, MD20742
{zzj,basili,ben}@cs.umd.edu

## Abstract

Inspection is a fundamental means of achieving software usability. Past research showed that the current usability inspection techniques were rather ineffective. We developed perspective-based usability inspection, which divides the large variety of usability issues along different perspectives and focuses each inspection session on one perspective. We conducted a controlled experiment to study its effectiveness, using a post-test only control group experimental design, with 24 professionals as subjects. The control group used heuristic evaluation, which is the most popular technique for usability inspection. The experimental design and the results are presented, which show that inspectors applying perspective-based inspection not only found more usability problems related to their assigned perspectives, but also found more overall problems. Perspective-based inspection was shown to be more effective for the aggregated results of multiple inspectors, finding about 30% more usability problems for 3 inspectors. A management implication of this study is that assigning inspectors more specific responsibilities leads to higher performance. Internal and external threats to validity are discussed to help better interpret the results and to guide future empirical studies.

## 1 Usability Inspection Techniques

Usability inspection [14] is an important approach to achieving usability. It asks human inspectors to detect usability problems in a user interface design so that they can be corrected to improve usability. It usually requires multiple inspectors, who can either work individually or as a team.

Usability inspection differs from user-based evaluation methods such as usability testing [6] or evaluation in participatory design [24]. In user-based methods usability problems are found through the observation of and interaction with users while they use or comment on an interface. In usability inspection, problems are found through the expertise of the inspectors and the inspection technique they use.

Different usability inspection techniques have been practiced, including heuristic evaluation, cognitive walkthrough, and formal usability inspection, etc. [20]. Empirical studies [5] [8] [11] showed that when using these techniques the percentage of usability problems detected by each inspector was rather low.

We use a list of internal and external attributes to characterize usability inspection techniques. Internal characteristics are those that are defined in the technique and are not supposed to be changed when the technique is used. The internal characteristics are:

**Prescriptiveness** This refers to the extent to which the technique guides the inspectors to do the inspection. This ranges from intuitive, non-systematic procedures to explicit and highly systematic procedures.

**Individual responsibility** Each inspector may be told to conduct the inspection in a general way, i.e., to identify as many problems as possible. Or each inspector may be assigned a specific role, i.e., to focus on a subset of issues at each moment.

**With or without meeting** A complete inspection may consist of individual inspections only, individual inspections followed by a meeting, or inspection meeting(s) only.

**Artifact coverage** Currently there are three approaches: (1) variety-based: have multiple inspectors explore the interface individually in a free way; (2) artifact-based: review each component (e.g. a dialog box or a Web page) of the interface; (3) task-based: define a set of representative user tasks and let the inspectors go through the tasks and at least check the components of the user interface encountered.

**Usability coverage** This refers to the usability issues that the inspection technique addresses. This can be part or all of *ease of learning, efficiency of use, retain over time, error handling, and user satisfaction* [20], with respect to different users and different working environment.

External characteristics are factors that are not defined by the technique but will be part of each instantiation of the technique and will have an influence on the effectiveness of problem detection. They include *computer support* [17], *artifact format* [10], *inspector expertise and characters* [14], and *organizational issues* [18].

The following usability inspection techniques have been empirically studied:

- *Cognitive Walkthrough* [23] inputs a description of the user interface, a set of task scenarios, assumptions about the knowledge a user will bring to the task, and the specific actions a user must perform to accomplish the task with the interface. The inspectors all meet together, led by a moderator, to examine each step in the correct action sequence by asking a set of predefined questions. It focuses on understanding and learning in novice use.
- *Guidelines or Standards Inspection* [13] is to have experts on some user interface guidelines or standards check the interface for compliance.
- *Heuristic Evaluation* involves having a set of evaluators examine the user interface and judge its compliance with recognized usability principles (the "heuristics"). Each individual evaluator inspects the system alone, using or not using task scenarios. Its effectiveness depends on the expertise of the inspectors and the variety of their inspections.

Based on 19 studies of heuristic evaluation, Nielsen [11] reported that on average each inspector could detect around 20%, 40%, or 60% of the usability problems depending on whether they were novices (with no expertise in either usability or the application domain), single-experts (with expertise in usability principles but without expertise in the specific application domain), or double-experts (with expertise in both usability and the application domain).

In a study conduced by Jeffries et al. [8] a team of 3 software engineers were able to find about 1/6 of the usability problems by using guidelines inspection. Among the problems found, only about 1/3 were found via the technique itself, with others found as a side effect (e.g., while applying a guideline about screen layout, a problem with menu organization might be noted) or through prior experience. These studies and two other studies [21] [22] show that it is not effective to simply use usability guidelines for either design or evaluation.

Desurvire [5] conducted a study where a phone-based interface was evaluated by groups of three evaluators of different experience levels. They used either heuristic evaluation or cognitive walkthrough. The three different experience levels were: experts who had at least three years of human factors work experience, non-experts who had less experience in usability, and software developers. The results showed that on average each non-expert inspector found about 8% of the problems while each software developers found about 16%, no matter which technique was used. But the expert group using heuristic evaluation did better than the expert group using cognitive walkthrough.

In summary, past research suggests that heuristic evaluation works fine when used by usability experts. But the current techniques are either not empirically studied, or shown to be ineffective for non-experts.

## 2 Perspective-based Usability Inspection

### 2.1 Introduction

Since it is difficult for each inspector to detect all different usability problems at the same time, we proposed perspective-based usability inspection [25], where each inspection session focuses on a subset of usability issues covered by one of several usability perspectives. Each perspective provides the inspector a point of view, a list of inspection questions that represent the usability issues to check, and a specific procedure for conducting the inspection. Our assumption is that with focused attention and a well-defined procedure, each inspection session can detect a higher percentage of the problems related to the perspective used, and that the combination of different perspectives can uncover more problems than the combination of the same number of inspection sessions using a general inspection technique.

This idea is supported by studies on defect-based and perspective-based reading of software requirement documents [1] [16]. These two studies showed that when inspecting requirement documents, it is more effective to let each inspector focus on one class of defects or inspect from one particular perspective than to let each inspector have the same and general responsibility.

Supportive evidences also came from a study by Desurvire [4], where each of the three levels of evaluators – human factors experts, non-experts, and developers – were asked to study flowcharts of a voice interface several times, once from each of several quite different perspectives. The perspectives used were of: the inspector's own, a human factors expert, a cognitive psychologist, a behaviorist, a Freudian, an anthropologist, a sociologist, a health advocate, a worried mother, and a spoiled child. All evaluators received the same order of perspectives. The results suggested that the perspectives approach may offer substantial promise as a technique to enhance inspection effectiveness by non-experts and developers. Unfortunately there was no discussion of why this list of perspectives was chosen and how efficient it was to use so many perspectives.

Kurosu et al. [9] developed "structured heuristic evaluation", where each usability session was divided into sub-sessions, with each sub-session focusing on one of the following: operability, cognitivity, pleasantness, novice/expert, and disabled users. They reported that their proposed method revealed more than twice the number of problems revealed by heuristic evaluation.

Sears [19] developed "heuristic walkthrough" by providing each inspector a prioritized list of user tasks, a list of usability heuristics, and a list of "thought-focusing" questions. The inspection is a two-pass process. Pass 1 is task-based exploration, guided by the list of thought-focusing questions. Pass 2 is free exploration, guided by usability heuristics. Inspectors detect usability problems in both passes. An empirical study found that heuristic walkthrough detected about the same number of usability problems as heuristic evaluation did, but reported much less false positives.

### 2.2 Overview of the Technique

In developing the technique, we first defined a model of human-computer interaction (HCI). Then we defined usability perspectives and the usability goals for each perspective. For each perspective, we went through the HCI model and generated questions about whether the relevant usability goals for that perspective can be achieved. This generated a list of generic usability questions for each perspective. Although these generic questions can be used in usability inspection, they can be tailored based on the characteristics of a certain kind of interfaces, such as the Web interfaces. Once such a tailoring is done for a certain type of interfaces, it can be used for all interfaces of that type. The tailored questions are more specific and relevant to the interface being inspected. For each perspective, the inspection questions are integrated into an inspection procedure for that particular perspective.

### 2.3 A Model of Human-Computer Interaction

In order to define the usability issues to inspect, we need to understand the human-computer interaction process. Our model extends Norman's "Seven Stages of Action" model [15] by adding error handling. The model characterizes a user's actions when using a computer by an iterations of the following steps: 1) Form the goal; 2) Form the intention; 3) Identify the action; 4) Execute the action; 5) Perceive the system response; 6) Interpret the results; 7) Understand the outcome; 8) Deal with errors that may have occurred.

Here a "goal" is a step towards accomplishing a task. For example, if the task is to fill out an on-line credit card application form, then a goal can be to fill out the name field, or to fill out the date-of-birth field, etc. A user needs to map such a goal to an action on the computer, execute the action, perceive and understand the feedback from the system, and examine if anything has gone wrong.

The iteration of these steps can also be summarized as cycles of *execution* (carry out some actions on the computer) and *evaluations* (judge how much the goal has been achieved and whether an error has been made), with possible error corrections. Therefore, the model naturally identifies two categories of usability problems: the *gulf of execution* (the mismatch between the user's intention and the allowable actions) and the *gulf of evaluation* (the mismatch between the system's representation and the user's expectations).

### 2.4 Usability Perspectives

Perspectives are used to focus the inspector's attention on a specific subset of usability issues during each inspection session. The perspectives should be as mutually exclusive as possible. The combination of different perspectives should cover all usability issues as much as possible.

Compared to the steps in the HCI model, the usability perspectives are higher-level scenarios of human-computer interaction. Different perspectives emphasize different stages in the HCI model, or different aspects of the same stage.

When using a computer to accomplish tasks, a user will experience one or more of the following situations:

**Novice use** The user's knowledge and experience do not tell the user how to use the system to achieve the goal.

**Expert use** The user knows how to use the system but prefers to achieve the goal efficiently and easily, or wants to achieve higher goals.

**Error handling** The user has a problem with the effect achieved by the previous action and needs to resolve the problem.

These three perspectives were defined based on the following two questions:

1. Whether or not the user knows how to achieve the goal;
2. Whether or not the user executes the action correctly.

If the answer to question 2 is "no", then the situation is covered by "error handling". Otherwise, answering "no" to question 1 leads to "novice use", and answering "yes" leads to "expert use". Therefore, both "novice use" and "expert use" only consider user actions along the correct path.

These three situations form the three perspectives we are using in the proposed usability inspection technique. Other perspectives may be used, especially for special application or user interface situations.

### 2.5 Usability Goals and Inspection Questions

Inspection questions are provided with each perspective to cover the usability issues to be examined. They are based on the HCI model and the perspectives. Along with the three perspectives, the following usability goals are defined:

**Novice use** The fundamental tasks can be accomplished by the defined users with the minimum knowledge.

**Expert use 1** Users can complete each task in an efficient and easy way.

**Expert use 2** Users can customize the system to behave the way they desire.

**Expert use 3** There are advanced functions or features that would enable expert users to be more productive.

**Error handling 1** The chances for user errors are minimized.

**Error handling 2** The user interface helps users understand the problem when user errors occur.

**Error handling 3** The user interface helps users recover from errors.

**Error handling 4** System failures are dealt with appropriately.

For each perspective, the inspection questions are generated by going through the steps in the HCI model and asking whether the usability goals for that perspective are achieved. In generating the questions, the characteristics of the user interface, the users, the tasks, and the users' working environment are considered.

## 2.6 Inspection Procedures

We provide inspectors an inspection procedure for each perspective. The procedure is designed to help inspectors organize the inspection process, so that the right usability issues are checked at the right time, and that the chance for neglecting some issues will be reduced.

For the "novice use" perspective, inspectors are asked to think about users who are not familiar with the interface and need guidance from the interface to find the correct action, to execute the action correctly, to notice the system feedback, and to understand the action results. Inspectors are asked to check for each task whether a novice user will successfully go through the above steps. Specific usability questions are organized under these steps.

For the "expert use" perspective, inspectors are asked to think about users who are familiar with the interface and to examine the interface for efficiency, flexibility, and consistency in supporting the user tasks, and check whether the interface has appropriate visual appearance and organization. The inspectors are asked to get familiar with the interface first. Then they are asked to go through the tasks. For each task, they should check if facilities such as short-cuts and default values are provided when possible, if the amount of hand or eye movement needed is minimized, etc. Each time a new screen shows up, the inspectors need to examine the colors, the fonts, and the organization of information on the screen based on the provided criteria.

For the "error handling" perspective, inspectors need to first derive the possible user errors and possible system failures for each task, based on a provided error classification. Then for each possible error, inspectors check to see if the interface has minimized the possibility for the error to occur; when the error occurs, if the interface provides informative error messages and minimizes the negative outcome of the error; and if the interface has sufficient support for users to recover from the error.

### 2.7 Summary

Perspective-based usability inspection is different from a general technique such as heuristic evaluation in two aspects. First it gives different inspectors different and focused responsibilities, as opposed to the same general responsibility. Second, it provides an inspection procedure for each perspective, as opposed to just a list of usability issues.

## 3 Method

### 3.1 Hypotheses

The hypotheses that were made before the experiment were:

- At the individual level, subjects using perspective-based inspection will detect a higher percentage of usability problems covered by their assigned perspective than subjects using heuristic evaluation.
- For the aggregation of multiple inspectors, perspective-based inspection will detect significantly more usability problems than heuristic evaluation.

### 3.2 Design of the Experiment

An experiment was conducted at a government organization. We had 24 professionals from the organization participated as subjects.

#### 3.2.1 The Constraints

This was an exploratory study in that we worked with the organization on a live project with limited resources. We could not set a significance value beforehand and calculate the statistical power to determine the number of subjects we were going to have. Instead we had to rely on the organization to find qualified volunteers to participate.

In addition, the project required the evaluation of two interfaces under equal conditions. Our experimental design had to balance the order that the two interfaces were inspected, and thus introduced the interface order independent variable.

Due to these constraints, we were not testing the hypotheses based on a predefined significance level. Rather we decided to use the 0.10 significance level in describing the results.

#### 3.2.2 The Design

We used a post-test only control group experimental design. The control group used heuristic evaluation. The experiment group was further divided into three sub-groups along the three perspectives.

Table 1: The number of subjects in each group

| | **Control group** | **Experiment group** | | |
|---|---|---|---|---|
| Interface order | Heuristic | Novice | Expert | Error |
| A, B | 6 | 2 | 2 | 2 |
| B, A | 6 | 2 | 2 | 2 |

Each subject was assigned to use one technique to inspect two alternative interfaces of a Web-based data collection form, namely interfaces A and B. The subjects were randomized and assigned to different techniques and different interface orders. The layout of the experimental design is shown in Table 1, where the numbers indicate the number of subjects in each treatment.

### 3.2.3 Factors in the Design

Based on the experimental design, the factors that were likely to have an impact on the results were inspection technique and interface order.

According to the framework defined in Section 1, the the two inspection techniques differ along the following dimensions:

**Prescriptiveness** Perspective-based inspection provides an inspection procedure for each perspective. Heuristic evaluation does not provide a procedure.

**Individual responsibility** Perspective-based inspection gives each inspector a focused responsibility. Heuristic evaluation gives each inspector the same and general responsibility.

**Artifact coverage** Perspective-based inspection emphasizes going through user tasks during the inspection. Heuristic evaluation does not require going through user tasks. However, for the Web-based forms being inspected in this experiment, the user task was very clear and straightforward. The whole interface was to support this user task. Therefore, this factor was not expected to have a significant impact on the results.

Other factors that may have an impact, but were not under control of the experimental design, are discussed in Section 5.

### 3.2.4 The Subjects

The 24 subjects in the experiment were familiar with both the interface domain and the task domain. They were either programmers, domain experts, technical researchers, or cognitive researchers. Efforts were made to evenly distribute participants of different backgrounds to different groups. But due to some schedule change, there were 5 programmers in the control group and 3 in the experiment group. The experiment group had 3 cognitive researchers while the control group had only 1. This imbalance will be discussed in the threats to validity.

### 3.2.5 Pilot Study and External Expert Reviews

Before the main study, we conducted a pilot study with 7 graduate computer science students to test out the instruments. We also asked two external usability experts to review the interfaces and report the usability problems they found. The problems they found were compiled into the list of detected problems. But the statistical analyses as presented in this paper only include subjects from the main study.

## 3.3 Experiment Procedure

In the main study, each subject first watched a video introduction of the project background and the inspection technique to be used. Then the subject was asked to sign a consent form and answer questions about previous experience in using and developing for the Web. After this, each subject spent up to 100 minutes in one of the two "cognitive lab" rooms to conduct the inspection. All inspection sessions were observed from an adjacent room through one-way mirrors. The sessions were also videotaped, with two views: one of the computer screen and the other of the inspector's facial expression and upper-body movement. Subjects were given forms for reporting detected usability problems. After the inspection, each subject was given a questionnaire form to fill out, which asked the subject to rate the ease of use of the technique, etc.

## 3.4 Materials

The usability heuristics and perspective-based inspection procedures used in the experiment are included in the appendix.

The usability heuristics used were:

1. Speak the users' language
2. Consistency
3. Minimize the users' memory load and fatigue
4. Flexibility and efficiency of use
5. Use visually functional design
6. Design for easy navigation
7. Validation checks
8. Facilitate data entry
9. Provide sufficient guidance

Each heuristic had a detailed explanation about the related usability issues.

For "novice use" perspective, inspectors were asked to think of novice users with a list of characteristics: being able to use a keyboard and a mouse, without visual disabilities, etc., which were defined based on the context of the application. Inspectors were given the description of the application and the user tasks. For each task, they were asked to think about whether a novice user would be able to choose the correct action, execute it successfully, and understand the outcome. They were provided with a list of detailed usability questions. For example, for data entry:

*Are formats for data entries indicated?*

For "expert use", inspectors were asked to think about expert users and check the interface for efficiency, flexibility, and consistency in supporting the user tasks. They were given a list of usability questions relating to these issues. For example, for data entry:

*Are possible short-cuts (e.g. using the Tab key to switch to the next field) available?*

*Are possible default values used?*

For "error handling", inspectors were given a classification of user errors. They were also given the characteristics of the users as the "novice use" inspectors were. For each user task, inspectors were asked to list the possible user errors and check the following questions for each user error:

*Does the user interface prevent the error as much as possible?*

*Does the user interface minimize the side effects the error may cause?*

*When the error occurs, will the user realize the error immediately and understand the nature of the error from the response of the user interface?*

*When the error occurs, does the user interface provide guidance for error recovery?*

## 3.5 Data Coding

**Step 1 A list of usability issues raised by each inspector**
After the experiment, we went through the usability report forms and built an accumulated list of detected usability issues for each interface. For each issue raised by an inspector, if it did not exist in the current list of issues, a unique number would be assigned. The issue would be added to the accumulated list under that unique number. The number would then be written down on the inspector's problem report form. If the same issue had been raised before, then just the number of that issue would be written on the appropriate place in the problem report form.

The same procedure was followed to process the usability issues raised by the 7 subjects in the pilot study and the 2 external expert reviewers.

In this way a list of usability issues raised by each inspector was obtained. A list of usability issues for each interface was also obtained. They were in the form of a list of numbers, with each number corresponding to a usability issue.

**Step 2 A list of all detected usability problems for each interface, with assigned severity levels**

Severity levels were assigned to the raised issues by a group of three people. The raters all had extensive experience in usability evaluation. Each person first rated the issues alone. Then meetings were held to go through each raised issue and determine its severity. If different severity levels were assigned to the same issue by different people, the difference would be resolved through discussions.

Nielsen's rating scale [12] was used to assign severity levels to the usability issues. The rating scale is as follows:

- 0 – This is not a usability problem.
- 1 – Cosmetic problem only, need not be fixed unless extra time is available.
- 2 – Minor usability problem, fixing this should be give low priority.
- 3 – Major usability problem, important to fix, so should be given high priority.
- 4 – Usability catastrophe, imperative to fix this before product can be released.

After this, the issues that had been assigned severity rating of 0 were removed from the list. In cases where two issues were recognized to be the same, it would be recorded that the corresponding two numbers referred to the same problem and one of them would be removed from the list. The final list of usability problems detected by each inspector was obtained by removing the ones that were not usability problems, changing the numbers of the ones that were removed from the overall list because they were the same as others in the list, and removing any duplicates.

The list of usability problems for each interface was obtained after removing from the accumulated list the duplicates and the ones that were regarded as not usability problems.

**Step 3 Usability problems under each category**
For the purpose of comparing the percentage of problems each inspector detected within responsibility, we went through every usability problem to see if it is covered by the heuristics, the novice use perspective, the expert use perspective, and the error handling perspective. If a problem is not covered by any of the above, it goes to the "other" category.

## 4 Results and Discussion

Altogether 82 problems were detected for interface A, 61 for interface B. These problems were collectively identified by the 24 experiment subjects, 7 pilot subjects, and 2 external expert reviewers.

The performance of the 24 experiment subjects is presented and discussed as follows.

### 4.1 Independent and Dependent Variables

The primary independent variable was the inspection technique. But another independent variable, the interface order, was introduced in the experimental design. Statistical tests failed to reveal a significant interaction effect between the inspection technique and the interface order, as shown in Table 2.

Table 2: Effect of independent variables on overall detection effectiveness (*p*-values from ANOVA)

| Source | Order | Technique | Order × Technique |
|---|---|---|---|
| Interface A | 0.50 | 0.19 | 0.23 |
| Interface B | 0.71 | 0.15 | 0.76 |
| Both A & B | 0.76 | 0.19 | 0.48 |

The dependent variables were the effectiveness of each inspection technique with respect to:

- the total number of usability problems detected, and
- the number of each class of usability problems detected.

The second effect is important as it tests the ability of each inspection technique to focus on a particular class of usability problems and suggests benefits for a team of inspectors using different perspectives.

## 4.2 Analyses

The statistical results were similar whether or not the severity ratings of the usability problems were considered. Therefore we are only presenting results when the severity ratings are not considered.

### 4.2.1 Individual Detection Effectiveness for All Problems

As stated before (Section 4.1), there were two independent variables: the interface order (interface A first or interface B first) and inspection technique (heuristic evaluation or perspective-based inspection). We used ANOVA to test the effect of each of these two variables as well as their interaction on the individual detection scores on interface A and on interface B. We used MANOVA to test these effects when the detection scores on the two interfaces by each inspector were considered at the same time. The detection score here is the number of usability problems detected.

The results of the ANOVA and MANOVA tests are shown in Table 2. It failed to reveal a significant effect by the interface order. For the inspection technique, there was also no significant effect shown (p=0.19 for interface A, p=0.15 for interface B, and p=0.19 for both). The interaction between inspection technique and interface order was found to be non-significant.

Another way to deal with the order effect is to compare the performance of the two techniques on the each interface when only the subjects who reviewed the interfaces in the same order are considered. Thus four t-tests were performed and the results are given in Table 3. It shows the average detection effectiveness in terms of the percentage of problems detected, as well as the *p*-values when the means from the two techniques are compared. In all cases, the perspective-based technique performed better than the heuristic technique, although in only one of the four situations there was a statistically significant difference (at 0.10 level). It should be noted that the sample size in each of these tests is 6 data points in each group, which is half of the subjects.

Table 3: Percentage of problems found for each interface-order situation

| Interface | Order | Heuristic | Perspective | $p$-value |
|---|---|---|---|---|
| A | A-B | 8.0 | 11.8 | 0.07 |
| A | B-A | 8.8 | 9.0 | 0.46 |
| B | A-B | 9.5 | 14.3 | 0.12 |
| B | B-A | 9.3 | 12.5 | 0.20 |

It is interesting to note that for interface A, the perspective-base technique performed much better than the heuristics technique when interface A was inspected first, while the two techniques performed almost the same on interface A when interface B was inspected first. Interface A was developed in HTML and had a "standard" look-and-feel that was familiar to all the subjects. Interface B was developed in Java and had an "ad hoc" interface that was much stranger to the subjects. Therefore, this may indicate that late in the inspection process, when the artifact being inspected was familiar to the inspector, the inspector may tend to ignore the inspection technique being used and fall back to his/her own way of doing the inspection. Thus the effect of the techniques tend to diminish in such situations.

As the evidence about how the subjects followed the assigned techniques, observation records and video-recordings show that most subjects read the instruction for the technique at the beginning. Some of them referred to it several times in the first 20 minutes. Almost nobody looked at the instruction again for the second interface. It is possible that they understood the general idea of the technique after a while. But it is unlikely that they had remembered the specific usability issues and the inspection procedures.

In summary, when data from all 24 subjects (with two independent variables) were considered, the inspection techniques did not have a significant effect on the detection of overall problems, as shown in Table 2. When only half of the subjects who reviewed the two interfaces in the same order were considered each time, for the subjects who reviewed interface A first, perspective-based inspection performed significantly better than heuristic evaluation (p=0.07). There was not a statistical significance for other situations.

But the perspective-based technique asks each inspector to focus on a subset of issues. Therefore each perspective inspector is not expected to find more overall problems. Our hypotheses were that individual perspective inspectors should find more problems related to their assigned perspectives, and that the combination of inspectors using different perspectives should be more effective than the combination of the same number of heuristic inspectors.

It is surprising that as shown in Table 3 perspective inspectors outperformed the heuristic inspectors at individual level for overall problems (although the differences were not statistically significant for 3 out of 4 cases). This is consistent with results from two other studies [2] [16] where inspectors with a focused responsibility detected more overall defects when reviewing software requirement documents.

Table 4: The effect of technique on the detection of problems by category ($p$-values from ANOVA)

| Category | Interface A | Interface B | Both |
|---|---|---|---|
| Novice | 0.065 | 0.043 | 0.044 |
| Expert | 0.42 | 0.29 | 0.40 |
| Error | 0.039 | 0.044 | 0.032 |

### 4.2.2 Individual Detection Effectiveness for Different Types of Problems

One hypothesis about the perspective-based technique was that compared to inspectors using heuristic evaluation,

- Inspectors using the "novice use" perspective would detect a significantly higher percentage of problems related to the "novice use" perspective.
- Inspectors using the "expert use" perspective would detect a significantly higher percentage of problems related to the "expert use" perspective.
- Inspectors using the "error handling" perspective would detect a significantly higher percentage of problems related to the "error handling" perspective.

First, ANOVA and MANOVA tests were run to test the effect of technique (four levels: heuristic and the three perspectives), interface order (two levels: interface A first or B first), and their interaction on the detection of problems covered by each perspective. Each test involves data from all 24 subjects. The interface order and the interaction between the technique and order were found to have no significant effect in any case. Table 4 shows the effect by technique. It shows a significant effect of inspection technique on the detection of "novice use" and "error handling" problems. The use of "expert use" perspective did not have a significant effect, possibly because that the inspectors themselves were all experts in the application domain and user interface domain. Thus they were able to capture a large portion of the "expert use" problems even without help from the "expert use" perspective.

Then 3 ANOVA tests were run between the heuristic group and each of the 3 perspectives, with both the technique and order variables considered. Each ANOVA involved data from 16 subjects (12 from heuristic evaluation and 4 from a perspective sub-group). Table 5 shows the results of these tests. For usability problems related to each perspective, the average percentage of such problems detected by the 4 inspectors using that perspective and the average percentage by the 12 heuristic inspectors are listed. The standard deviations are in parentheses. It shows that the use of the "novice use" and "error handling" perspectives significantly improved the inspector's detection effectiveness for problems related to the perspectives.

In summary, the results of this analysis supported the hypotheses for both "novice use" and "error handling" perspectives. The "novice use" inspectors found significantly more problems related to novice us than the heuristic inspectors. The "error handling" inspectors found significantly more problems related to user errors than the heuristic inspectors. But there was not a statistically significant difference for the "expert use" perspective. A possible reason of this was given in the above discussion.

Table 5: Comparison of different types of problems found

| | Category | % of problems by 12 heuristic subjects | % of problems by 4 perspective subjects | $p$-value |
|---|---|---|---|---|
| A | Novice | 8.0 (6.6) | 18.5 (9.0) | 0.025 |
| | Expert | 15.9 (10.3) | 20.5 (8.7) | 0.477 |
| | Error | 14.3 (12.2) | 33.9 (6.8) | 0.012 |
| B | Novice | 11.7 (9.1) | 26.3 (11.1) | 0.019 |
| | Expert | 14.9 (11.1) | 28.0 (18.5) | 0.134 |
| | Error | 9.0 (10.9) | 29.3 (13.5) | 0.013 |

(standard deviations are in parentheses)

Table 6: Correlation between experience and inspection performance

| | Experience using the Web | Experience developing for the Web |
|---|---|---|
| Problems for A | -0.416 | 0.010 |
| Problems for B | -0.187 | 0.194 |
| Time for A | -0.175 | 0.067 |
| Time for B | 0.339 | 0.316 |

### 4.2.3 Correlation between Experience and Performance

Subjects were asked to give a self-assessment of their own Web use skills on a 1 to 9 scale. Subjects were also asked how many Web sites they had developed, with 3 options: none, a few, or many. Table 6 shows the correlation coefficients of these two measures and the number of problems found as well as the time spent doing the inspection. There was no strong correlation between experience and inspection performance.

### 4.2.4 Aggregation of 3 Inspectors

Although all inspectors conducted the inspection individually, we were interested in comparing the aggregated results of multiple inspectors. For example, we compared the number of unique problems identified by 3 perspective inspectors (one from each of the three perspectives) and 3 heuristic inspectors (any 3). There were 220 possible aggregations for heuristic evaluation and 64 for perspective-based inspection. Table 7 shows the average performance of all such possible aggregations for each technique group. Since the data points under each group were not independent from each other, no statistical test was performed.

### 4.2.5 Permutation Test of All Possible 12-person Aggregations

We did a permutation test [7] of simulated 12-person teams. This involves constructing all possible 12-person teams and see how the un-diluted perspective team ranked among all possible 12-person

Table 7: Aggregated problems found by 3 inspectors

| Interface | Technique | % of problems found | Improvement |
|---|---|---|---|
| A | Heuristic | 21.8(5.0) | |
| | Perspective | 27.7(4.4) | 26.5% |
| B | Heuristic | 24.1(7.2) | |
| | Perspective | 32.8(7.4) | 35.7% |

(standard deviations are in parentheses)

Table 8: Permutation tests for all possible simulated 12-person teams

| | Number of possible teams | Rank of the perspective team | $p$-value |
|---|---|---|---|
| A | 2,704,156 | 262,577 | 0.097 |
| B | 2,704,156 | 122,993 | 0.045 |

teams in terms of number of unique problems detected. Whether or not we can claim that the perspective-based technique had a beneficial effect on team performance depends on how the un-diluted perspective team (with all 12 perspective inspectors) appears towards the top of the ranking. The $p$-value is the rank of the un-diluted team divided by the total number of teams. There were 2,704,156 possible 12-person teams out of the 24 subjects. The results of this test are given in Table 8. It shows that at $p < 0.10$ level, the perspective-based inspection technique significantly improved the effectiveness of an inspection team.

#### 4.2.6 The Overlapping among Problems Detected by Perspective Sub-groups

This analysis looked into the overlapping of problems detected by each perspective sub-group. As shown in Figure 1, the number in a circle slice represents the number of usability problems uniquely detected by the combination of 1, 2, or 3 perspective sub-groups, depending on whether the circle slice is occupied by 1, 2, or 3 full circles of the three perspectives. For example, for interface B, there were 6 problems that were detected by all three perspectives, 4 detected by novice and error perspectives but not by expert perspective, and 15 detected by novice perspective alone. Although there is no other data to compare against at the moment, it shows that for both interfaces the different perspective sub-groups detected fairly different usability problems.

#### 4.2.7 Major Problems Detected Only by One Technique Group

In this analysis, we went through all the detected problems that were ranked 3 (major usability problem) or 4 (usability catastrophe) and counted how many unique problems were detected by only one of the two technique groups (the control group and the experiment group). For interface A, heuristic inspectors (control group) did not find any unique problems, while perspective inspectors (experiment group) detected 9 unique problems by a total of 16 times (i.e. some problems were

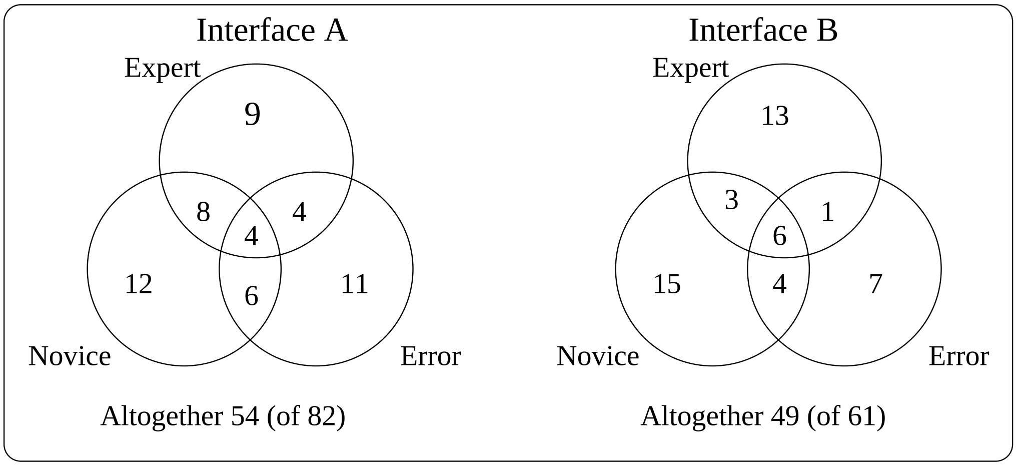

Figure 1: Overlapping of problems detected by different perspectives

detected by more than one perspective inspectors). Of these 9 problems, 4 were detected by only one inspector; 3 were detected by two inspectors; and 2 were detected by three inspectors. For interface B, each technique group detected 4 unique problems. But each of the 8 unique problems were only detected by one inspector. This shows that giving inspectors specific responsibilities did not make them less effective in detecting major usability problems.

# 5 Threats to validity

Threats to validity [3] are factors other than the independent variables that can affect the dependent variables. Such factors and their possible influence are discussed in this section.

## 5.1 Threats to Internal Validity

The following threats to internal validity are discussed in order to reveal their potential interference with the results:

- **Maturation:** In this experiment, the whole inspection took up to 1 hour and 40 minutes, with no break. The likely effect would be that towards the end of the inspection session, the inspector would tend to be tired and perform worse. Also since the two interfaces had the same content, it is likely that for the second interface inspected, the inspector got bored and did not do the inspection as thorough as before. However, from observation records of the experiment, there were no sign showing that the subjects looked tired or bored. The experimental design let half of the subjects inspect interface A first while the other half inspect interface B first. The two interfaces differed to a large extent in terms of look and feel, which helped to keep the subjects interested. An ANOVA test failed to show a significant effect of the order on individual performance.

- **Testing:** Getting familiar with the material and the technique may have an effect on subsequent results. This is potentially a threat to this experiment since each subject used the same technique for both interfaces, and that the two interfaces had the same content with different presentations. The experimental design had exactly the same number of subjects within each technique group inspect the two interfaces in two different orders. This should counter-balance some of the effect both between the two groups and within each group.
- **Instrumentation:** In this experiment, the decisions in data coding were made by a group of three people through meetings and discussions. These decisions included whether an issue raised by an inspector is a usability problem, what severity level should be assigned to each problem, and whether a particular problem is covered by the heuristics and any of the three perspectives. It might be better to have each person do it separately, and to have meetings to see how consistent they are and to resolve the differences.
- **Selection:** As stated before, we tried to balance the number of subjects of different job background between the control group and the experiment group. The number of domain experts and technical researchers were balanced between the two groups. But due to some unexpected schedule change, the control group had 5 programmers and 1 cognitive researcher. The experiment group had 3 programmers and 3 cognitive researchers. This imbalance may have contributed to the differences between the two groups.
- **Process conformance** Another threat is that people may have followed the techniques poorly. For heuristic evaluation, the introduction video read through all the heuristics and the related usability issue of each heuristic. The inspectors had these heuristics and issues with them during the inspection. For perspective-based inspection, the introduction video described the idea of doing inspection from three different perspectives and mentioned briefly the usability issues under each perspective. Almost all subjects in the perspective group read through the provided instruction thoroughly before the inspection. But some subjects in the perspective group reported that they did not follow the technique well or could not follow the technique since “it would take too long” or “I don’t fully understand it”. Given the 2-hour limitation we were not able to provide better training and make sure all subjects understood and felt comfortable with applying the technique. Also the inspection procedure for each perspective as given in this experiment appeared to be too detailed and somewhat intimidating. Given the time limitation, it may have become not practical to literally follow the procedure. But we believe all subjects in the perspective group got the general idea about the perspectives and the usability issues. Most of them tried to follow the technique and focus on the assigned perspective. The different techniques asked different inspectors to conduct the inspection in different ways. If the process conformance had been better, the differences between the different technique groups should be larger, and thus achieving better experimental results.

### 5.2 Threats to External Validity

One possible threat to external validity is:

- **Reactive effects of experimental arrangements.** In this experiment, we did not tell the subjects that we were comparing two inspection techniques. The subjects only knew that they were supposed to use the assigned technique to detect as many usability problems as they could. We asked the subjects not to discuss with other subjects what they have done during the inspection before all subjects had finished participating in the experiment. Our impression was that the subjects were more interested in finding usability problems than using the techniques. The lab environment kept them concentrated on the inspection without distraction or interruption. The awareness that they were observed by others and video recorded may have some impact on their behavior. But since all these apply to both technique groups in the same way, they might not make a significant difference on the relative performance of the two techniques.

## 6 Conclusions and Future Directions

This experiment with 24 professionals found significant improvement in finding usability problems in a web-based application when a perspective-based inspection was used, as compared to a heuristic inspection. The improvement was approximately 30% for the aggregated results of 3 inspectors (Table 7). As predicted, perspective inspectors (novice, expert, error) found 30% to 3 times more usability problems related to the assigned perspective (Table 5) than the heuristic inspectors. Furthermore, the average number of all the problems found by each perspective inspector was also higher than that of each heuristic inspector (Table 3). Some of the results are shown in Figure 2. A management implication of this study is that assigning inspectors more specific responsibilities leads to higher performance. Combining multiple perspective inspections is a wise strategy for creating high quality user interfaces.

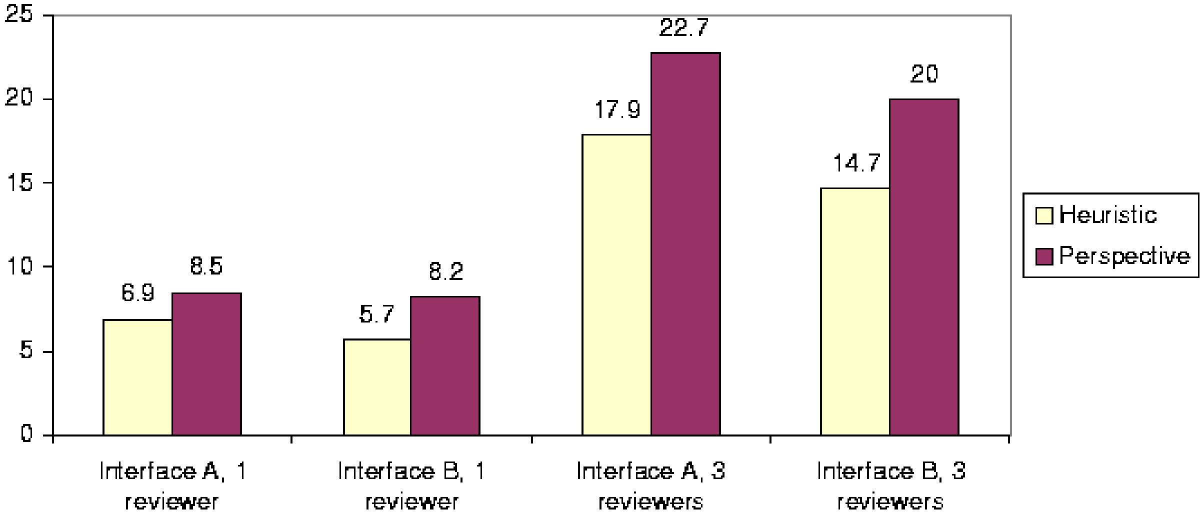

Figure 2: The number of detected problems at the individual level and for 3-reviewer aggregations

To deal with some of the threats to validity, more experiments are going to be conducted to see if the positive results can be replicated when some of the threats are removed.

To generalize the results, the following issues need to be considered:

- Domain experts vs. usability experts. The subjects in this experiment were all experts in the application domain, with some knowledge in usability. We need to know how the technique works for inspectors who are usability experts with some knowledge in the application domain, as well as for inspectors who are experts in both usability and the application domain.
- Inspection time. In the experiment, each inspector was given a time limit of 100 minutes to inspect the two interfaces. Although most participants finished the inspection within the time limit, there was one case where a perspective inspector said that given the time limit she was not able to follow the technique very well. In some studies, inspectors were asked to conduct the inspection, besides doing their daily work, within two weeks. It would be interesting to test how many more problems the inspectors can detect when they are given more time. Also if each subject has much more time, we may want to let each perspective inspector try out all the perspectives, one at a time. In practice, an inspector often has enough time to go through an interface several times in doing the inspection.
- Experience with the technique. In this experiment, both techniques were new to the subjects. We would like to know how the inspectors perform with more experience.

We plan to conduct more empirical studies to address some of these issues. A lab package is being built to facilitate replications of the experiment by other researchers. We also plan to build an application package so that practitioners can learn and use the technique and provide some feedback.

## 7 Acknowledgments

This research has been supported by SOGEI S.p.A. (Rome, Italy) and by the National Science Foundation under grant EIA 01-5-23394. We thank Elizabeth Nichols, Barbara Sedivi, Nancy VanDerveer, Ellen Soper, and Al Paez of the US Bureau of the Census for their cooperation in the study. We thank all the subjects for their participation. We thank Pawan Vora of US WEST and Chris North of University of Maryland for their expert reviews of the interfaces. We thank the anonymous reviewers for these thoughtful comments. We thank Chip Denman of University of Maryland for his help on statistics.

## References

[1] Victor Basili. Evolving and packaging reading technologies. *Journal of Systems and Software*, 38:3–12, 1997.

[2] Victor Basili, Scott Green, Oliver Laitenberger, Forrest Shull, Sivert Sorumgard, and Marvin Zelkowitz. The empirical investigation of perspective-based reading. *Empirical Software Engineering*, 1(2):133–164, 1996.

[3] Donald T. Campbell and Julian C. Stanley. *Experimental and Quasi-Experimental Designs for Research.* Houghton Mifflin Company, 1966.

[4] Heather W. Desurvire. Faster, cheaper!! Are usability inspection methods as effective as empirical testing? In Jakob Nielsen and Robert L. Mack, editors, *Usability Inspection Methods*, chapter 7, pages 173–202. John Wiley & Sons, Inc., 1994.

[5] Heather W. Desurvire, Jim M. Kondziela, and Michael E. Atwood. What is gained and lost when using evaluation methods other than empirical testing. In A. Monk, D. Diaper, and M. D. Harrison, editors, *People and Computers VII*, pages 89–102. Cambridge University Press, 1992.

[6] Joseph S. Dumas and Janice C. Redish. *A Practical Guide to Usability Testing.* Ablex Publishing Corporation, Norwood, New Jersey, 1993.

[7] E.S. Edington. *Randomization Tests.* Marcel Dekker Inc., New York, NY, 1987.

[8] R. Jeffries, J. R. Miller, C. Wharton, and K. M. Uyeda. User interface evaluation in the real world: A comparison of four methods. In *ACM CHI'91 Conference Proceedings*, pages 261–266. ACM, 1991.

[9] Masaaki Kurosu, Masamori Sugizaki, and Sachiyo Matsuura. Structured heuristic evaluation. In *Proceedings of the Usability Professionals' Association Conference*, pages 3–5, June 1998.

[10] Jakob Nielsen. Paper versus computer implementations as mockup scenarios for heuristic evaluation. In D. Diaper et al., editor, *Human-Computer Interaction – INTERACT'90*, pages 315–320. IFIP, 1990.

[11] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *ACM CHI'92 Conference Proceedings*, pages 373–380. ACM, 1992.

[12] Jakob Nielsen. *Usability Engineering.* Academic Press, Inc., San Diego, California, 1993.

[13] Jakob Nielsen. Heuristic evaluation. In Jakob Nielsen and Robert Mack, editors, *Usability Inspection Methods*, chapter 2, pages 25–62. John Wiley, 1994.

[14] Jakob Nielsen and Robert Mack, editors. *Usability Inspection Methods.* John Wiley & Sons, Inc., 1994.

[15] Donald A. Norman. *The Design of Everyday Things.* Basic Books, New York, 1st doubleday/currency edition, 1988.

[16] Adam A. Porter, Lawrence G. Votta, Jr., and Victor R. Basili. Comparing detection methods for software requirements inspections: A replicated experiment. *IEEE Transactions on Software Engineering*, 21(6):563–575, June 1995.

[17] John Rieman, Susan Divies, D. Charles Hair, and Mary Esemplare. An automated cognitive walkthrough. In *ACM CHI'91 Conference Proceedings*, pages 427–428, 1991.

[18] Carolyn B. Seaman and Victor R. Basili. An empirical study of communication in code inspections. In *Proceedings of the 19th International Conference on Software Engineering*, pages 96–106, May 1997.

[19] Andrew Sears. Heuristic walkthroughs: Finding the problems without the noise. *International Journal of Human Computer Studies*, 9(3):213–234, 1997.

[20] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley, 3rd edition, 1998.

[21] Linda Tetzlaff and David R. Schwartz. The use of guidelines in interface design. In *ACM CHI'91 Conference Proceedings*, pages 329–333. ACM, April 1991.

[22] Henirik Thovtrup and Jakob Nielsen. Assessing the usability of a user interface standard. In *ACM CHI'91 Conference Proceedings*, pages 335–341. ACM, April 1991.

[23] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. The cognitive walkthrough method: A practitioner's guide. In Jakob Nielsen and Robert L. Mack, editors, *Usability Inspection Methods*, chapter 5, pages 105–140. John Wiley & Sons, Inc., 1994.

[24] Peter C. Wright and Andrew F. MonK. A cost-effective evaluation method for use by designers. *International Journal of Man-Machine Studies*, 35:891–912, 1991.

[25] Zhijun Zhang, Victor Basili, and Ben Shneiderman. Perspective-based usability inspection. In *Proceedings of the Usability Professionals' Association Conference*, pages 281–282, June 1998.

# A THE INSPECTION TECHNIQUES

The inspection techniques as used in the experiment are presented here.

## A.1 Usability Heuristics

1. **Speak the user's language**: Use words, phrases, and concepts familiar to the user. Present information in a natural and logical order. Define new concepts the first time they are used.
2. **Consistency**: Indicate similar concepts through identical terminology and graphics. Create consistent interfaces for tasks that are essentially the same. Adhere to uniform conventions for layout, formatting, phrasing, interface controls, task actions, etc.,for tasks that closely resemble one another.
3. **Minimize the users' memory load and fatigue**: Take advantage of recognition rather than recall. Do not force users to remember key information across tasks. Minimize physical actions such as hand movements, and mental actions such as visual search or decisions.
4. **Flexibility and efficiency of use**: Accommodate a range of user sophistication. For example, guide novice users through a series of progressive steps leading to the desired goal, but provide proficient users with shortcuts that do not violate data collection procedures.
5. **Use visually functional design**: Visually structure the user's task. Support frequent repetition of a small set of well specified tasks. Make it hard to confuse different tasks. User's eyes should be drawn to the correct place at the correct time, e.g. to actions to be performed, items to be remembered or referred to.
6. **Design for easy navigation**: Allow the user to move as necessary through the form, either forward or back to an earlier question. Enable an easy return from a temporary excursion to another portion of the survey. Enable user to determine current position easily.
7. **Validation checks**: Make sure error messages are clear. Resolution is easy. Placement of edit validations makes sense. Error validations will be performed.
8. **Facilitate data entry**: Easy to enter data. Data are visible and clearly displayed. Allow the users to change data previously entered. Easy to find data already entered. Necessary entries are clearly defined. Entries are in correct format.

9. **Provide sufficient guidance**: Convey sufficient text or graphical information for the user to understand the task, but do not provide more information than users need. Implicitly convey task instructions where possible through non-verbal cues, such as those provided by the spatial relationships among form elements on the screen. Provide help when necessary, either auditory or on-line.

## A.2 Inspection Procedure for Novice Use

The user's goal is to fill out the form and submit it. The goal can be decomposed into a series of sub-goals. For each sub-goal, go through the following stages and check the questions for each stage.

1. **Map** the **sub-goal to** the **effects** to be achieved in the user interface.
   (a) Will the user know when the subgoal is achieved?
2. **Identify** the **actions** for achieving the effects.
   (a) Are there *instructions or online help* that are *understandable* to the user and provide *sufficient guidance* as to what actions to execute?
   (b) Does the user know how to get to the online help, and how to come back from the online help?
   (c) Are *visual or auditory cues* like labels, icons and sound *understandable* to the user, and *consistent* from place to place in the user interface?
   (d) Do buttons and other clickable objects look clickable?
   (e) Are items in a list *unambiguous* in meaning?
3. **Execute** the **actions**. For each action
   (a) Are there *instructions or online help* that are *understandable* to the user and provide *sufficient guidance* as to how to execute the action (selection, data entry, navigation, submission, etc.)?
   (b) Can the user refer to the online help while answering questions?
   (c) Can the user execute the action correctly based on his/her previous knowledge?
   (d) Are same *actions* executed in a *consistent* way *among* different places in *the user interface*?
   (e) Are formats for data entry indicated?
4. **Perceive** the system **feedback**.
   (a) Does each user action (selection, data entry, navigation, submission, etc.) generate *feedback* that the *user is not likely to miss*?
   (b) Can users with disabilities or insufficient computer support (as described in the user profile) perceive the feedback?
5. **Understand** the **progress** made.
   (a) After each user action (selection, data entry, navigation, submission, etc.), will the *feedback* from the user interface *help the user* to *understand* if *progress* has been made?
   (b) Can the user constantly see what has been achieved so far?

## A.3 Inspection Procedure for Expert Use

The user's goal is to fill out the form and submit it. The goal can be decomposed into a series of sub-goals. For each sub-goal, go through the following stages and check the questions for each stage.

1. **Scan through the instructions, objects, and actions** in the user interface.
   (a) Is the text *easy to read*?
   (b) Is the information organized in a way that the most *important information can be read first*?
   (c) Is each list presented in a way that the *more frequently selected items appear earlier*?
   (d) Is *redundant information avoided*?
2. **Execute the actions** for achieving the sub-goal, using short-cuts whenever possible. For each action,
   (a) Are possible *short-cuts available*, e.g., allowing users to use keyboard to switch to the next text field?
   (b) Are possible *default values used*?
   (c) Does the system *do computation or remember information for the user* whenever possible?
   (d) Can the user *make a selection by clicking on a larger area* associated with the object to be selected, e.g., by clicking on the text next to the radio button to be selected?
   (e) Are *unproductive activities minimized*? These include navigation, mouse movements, hand movements between the mouse and the keyboard, and eye movements, etc.
   (f) Are *stressful actions minimized*? These include keeping a mouse button pressed for a long time, clicking a mouse button multiple times consecutively, using the mouse to click on a very small object.
3. **Wait for system response** if necessary.
   (a) Does each user action *immediately* generate *perceivable results* in the user interface?

Besides the above detailed inspection, you should also consider the following higher-level question:

- Can the structure of the Web-based form be re-designed somehow to significantly reduce the user's unproductive activities (navigation, mouse movement, hand movement between the mouse and keyboard, and eye movement, etc.)?

## A.4 Inspection Procedure for Error Handling

User errors often occur during human-computer interaction. The possible user error situations include, but not limited to:

- **Omission**: the user forgot to answer one or more questions; forgot to submit the form; etc.
- **Slippage**: the user typed something wrong; selected the wrong item or executed the wrong action (e.g. RESET) by accident; etc.

- **Wrong perception**: the user did not see a full list of possible answers because some items are not visible on the screen, or there is a visual break; etc.
- **Failed trial**: the user's guess turned out to be wrong. A novice user may guess on the basic functions, while an expert user may guess on short-cuts, etc.
- **Wrong system mode**: the user executed an action at the wrong mode (e.g. typing before activating a text field); entered data at the wrong location; navigated to the wrong place; etc.

With this Web-based form, a user's goal is to fill out the form with the complete and correct information and submit the form. This goal can be achieved by a series of steps. For each step of the user, go through the relevant parts of the user interface and consider all possible user errors that may occur. For each such error, ask the following questions:

1. Has the user interface done its best to *prevent the error*? (prevention)
2. When the error occurs, will the *user realize the error* immediately and *understand* the nature of *the error* from the response of the user interface? (information)
3. Does the user interface *minimize the side effects* the error may cause? (correction)
4. When the error occurs, does the user interface *provide guidance for error recovery*, including guidance about how to reverse the side effects? (correction)

Whenever the answer to one of the above questions is "no", a usability problem is detected. You may also detect problems not covered by these questions, but please make sure that you **focus on error handling issues as much as possible**.