McDonald, S. Cockton, G and Irons A. The impact of thinking aloud on usability inspection. Accepted (May 2020)

EICS 2020 published in ACM PACMHCI

The impact of Thinking-Aloud on usability inspection

Sharon McDonald

Faculty of Technology, University of Sunderland, Sunderland, United Kingdom, Sharon.McDonald@sunderland.ac.uk

Gilbert Cockton

Faculty of Technology, University of Sunderland, Sunderland, United Kingdom, Gilbert.Cockton@sunderland.ac.uk

Alastair Irons

Faculty of Technology, University of Sunderland, Sunderland, United Kingdom, [Alastair.Irons@sunderland.ac.uk](mailto:Alastair.Irons@sunderland.ac.uk)

ABSTRACT

This study compared the results of a usability inspection conducted under two separate conditions: An explicit concurrent think-aloud that required explanations and silent working. 12 student analysts inspected two travel websites thinking-aloud and working in silence to produce a set of problem predictions. Overall, the silent working condition produced more initial predictions, but the think-aloud condition yielded a greater proportion of accurate predictions as revealed by falsification testing. The analysts used a range of problem discovery methods with system searching being favoured by the silent working condition and the more active, goal playing discovery method in the think-aloud condition. Thinking-aloud was also associated with a broader spread of knowledge resources.

CCS CONCEPTS

• Human-centered computing~Human computer interaction (HCI)~HCI design and evaluation methods~Heuristic evaluations

KEYWORDS

Usability Inspection,Heuristic Evaluation,Think-aloud,Evaluation Resources

1 Introduction

Research into the use of think-aloud methods within usability testing has sought to isolate the elicitation conditions under which the concurrent think-aloud causes *reactivity*: an artificial, and often positive, change in task performance [[8](#bib8), [20](#bib20),[21](#bib21)]. These elicitation processes include asking for verbalisations that activate higher-order cognitive processes such as elaboration and explanation through focused instructions or interactive prompts [[5](#bib5), [7](#bib7)]. In usability testing, a reactive think-aloud is a cause for concern because it threatens the validity of the resultant data; but are there any circumstances in the general arena of evaluation methods, where reactivity might be a force for good?

In this paper, we report the results of transferring think aloud resources from usability testing to usability inspection. Specifically, we investigate whether asking inspection analysts to think-aloud can improve their performance by: helping them to adopt more active problem discovery strategies; and activating a range of knowledge resources to help them in problem analysis. We report on the first application of think-aloud within inspection. Possible benefits include improved predictions, learning that improves future inspections, and better exploitation of personal knowledge resources, which could have positive implications for both HCI education and UX Practice.

1.1 Evaluation in Interaction Design

Evaluation work in Interaction Design has been supported by empirical and analytical methods since the first years of HCI. ‘Method’ can create expectations of something complete, perhaps even an algorithm, that unambiguously and clearly guides evaluators through each step of a procedure. Quick consideration of empirical evaluation shows that such expectations will not be met. Approaches to usability testing are more of an agenda [[18](#bib18)], outlining the steps that need to be taken: profile and recruit test participants; design study; run study; analyse data; present results. For each of these ‘steps’, evaluators must make many independent choices. In settings with a high degree of usability maturity, many resources will be in place already: participant panels, screening questionnaires, informed consent forms, previous study plans, physical facilities and technology for running studies; statistics software; and previous reports. While naïve expectations about ‘methods’ are unrealistic, there is clearly much re-use in evaluation work, especially within organisations, but also as archived guidance and recommendation on inspection and testing.

A more accurate view of re-usable support for evaluation work was proposed by [[29](#bib29)]. Instead of complete methods, we have re-usable *approaches* (such as usability testing) supported by re-usable *resources* (such as think aloud). Both approaches and resources are typically either incomplete, need to be adapted, or both. For example, Heuristic Evaluation has no standard procedure for conducting an inspection (but some have been proposed [[4](#bib4)]), and heuristic sets may be extended and merged for specific projects. The need to adapt, complete, and sometimes add approaches or resources is reflected in the evaluator effect [[9](#bib9)], where different evaluators produce different results [[22](#bib22)], even from the same data [[9](#bib9)]. The evaluator effect can be explained in terms of having different resources, especially ones associated with individual or organisational differences. The impact of variations on evaluation performance has been shown for a range of resources, e.g.: task specifications for walkthrough [[25](#bib25)], problem report formats [[3](#bib3)], problem merging [[13](#bib13)] and think aloud [[21](#bib21)].

Methods are thus achievements. Evaluators must work to get methods to work by filling gaps in approaches and resources, with further modifications to fit project contexts. We have also seen steady moves away from overly simple assumptions about development processes, where each stage in a fixed sequence is supported by a specific set of methods. However, progress is slow in some areas. The updated ISO 9241-210 standard [[16](#bib16)] associates personas and ‘as-is’ scenarios with its first core stage of *Understanding and Specifying the Context of Use,* with which Contextual Design [[10](#bib10)] is also associated. However, with adaptation, a method can be usefully moved between stages. For example, [[23](#bib23)] adapted Contextual Design for evaluation work (the last core stage of ISO-9241-210).

Approaches such as Contextual Design have thus been successfully transferred from their initial intended development stage. There is less evidence of this being attempted with resources, i.e., to transfer a resource between approaches. Some resources have always been common to a few approaches, especially usability problem report formats [[19](#bib19)]. Most resource focused research however has studied the impact of alternative resources (as above, task specifications, problem merging, think aloud). Studies have typically been pragmatic, with no theoretical basis. In this study, we built on psychological theories of think-aloud elicitation from usability testing and apply think-aloud elicitation to usability inspection.

1.2 Think Aloud studies in Usability Evaluation

Think-aloud techniques have been used as a resource in usability testing for many years, and as a tool within psychology studies for much longer. While one domain has transferred resources to another, the context of use for think-aloud studies within usability testing has brought about changes in the elicitation process that may undermine the validity of the test. In its purest form, the concurrent think-aloud lays bare an individual’s thought processes as they perform a given task. The protocols therefore should relate predominantly to the content of working memory. According to meta-analytic studies and a large-scale review [[5](#bib5), [7](#bib7)] eliciting the think aloud using a neutral instruction, neutral think-aloud practice and prompts that are limited to “keep talking” should preserve the validly of the data. A think-aloud elicited in this way can be considered to be a classic concurrent think-aloud [[5](#bib5)]. However, when different elicitation processes are used, such as interactive or questioning prompts, reactivity can result [[5](#bib5)]. Reactivity is an artificial change in task performance. This change is related to the activation of higher-order cognitive process including reflection and elaboration which, therefore, go beyond the content of task-based working-memory processes. It may cause users to re-evaluate or reconsider their approach yielding a result that, otherwise, would not have occurred.

Within studies of think-aloud methods, reactivity is usually revealed by an increase in the number of correctly solved tasks compared to silent working [[7](#bib7)]. However, a modified think-aloud (focused instructions and/or prompts) may also lead to other behavioral changes without necessarily changing overall performance, these might be considered to be the precursors of reactivity. For example, a relaxed think-aloud using evaluator probes increased scrolling and link traversal [[8](#bib8)]; an explicit instruction requesting likes and dislikes increased within and between page navigation and scroll bar use [[20](#bib20)]. Moreover, some studies show increased problem yields (but without a concomitant increase in the severity of the problems extracted), through the use of explicit instruction requesting expectations, likes, dislikes and explanations [[32](#bib32)] and evaluator probes [[21](#bib21)]. Given that reactivity within usability testing is an artificial enhancement in test user performance, it should be avoided to preserve the validity of usability test data. However, when applied to a different stakeholder in the usability evaluation process could this same threat become an opportunity?

1.3 Thinking aloud about Usability Inspection

As noted above Usability inspection, as with other approaches, is subject to an evaluator effect. Different evaluators using the same approach, applied to the same product will yield different results. This may be explained by differences in the activities used during inspection and the knowledge and experience that evaluators bring to bear on the inspection process. Indeed, the finding that novice evaluators find fewer problems than experts [[17](#bib17)] may have less to do with their knowledge of the “approach’ and more to do with the general experience and knowledge they possess. The DARe model [[2](#bib2)] articulates a number of problem discovery strategies that may be used during inspection (scanning, system searching, goal playing and method following). The model also specifies a range of knowledge resources (e.g. task, user, product knowledge) that analysts may use to help them to reason about discovered problems. Techniques that help analysts activate these resources and apply them appropriately may help improve inspections particularly for novice analysts.

Instructional research has long since noted the benefits of asking learners to explain their understanding during problem solving and learning in a variety of domains [[1](#bib1), [28](#bib28)]. We suggest that the activities analysts engage in during inspection are analogous to the process of learning, where the learner (or analyst) makes a deliberate attempt to use their pre-existing knowledge to interpret new material and in turn, re-evaluate their own ideas in light of the new material during analysis where they confirm or eliminate the prediction. Therefore, techniques such as a modified think-aloud that solicit explanations from analysts during inspection may improve performance in terms of problem discovery in that it may yield changes in strategy or in terms of improving the accuracy of predictions made. Indeed, extended structured report formats were demonstrated to improve problem analysis [[2](#bib2)] but could think-aloud improve problem discovery too? Generating explanations during learning is thought to actively fill gaps knowledge by making better use of pre-existing knowledge [[28](#bib28)]. A modified think-aloud that requests explanations may have a similar impact during inspect by activating analysts’ existing knowledge resources to fill gaps within a method. For example, this analyst knowledge could support consideration of users, their tasks, the domain or the technology. This should help analysts to reflect on their activities during the process of problem discovery (in-action) where change in the activity is still possible [[26](#bib26)] rather than during post-discovery reflection (on action) when reviewing problem reports and where no further discovery is planned [[26](#bib26)].

In this paper, we report the results of an exploratory study in transferring think aloud resources from usability testing to usability inspection. We are interested in the transfer of a particular type of think-aloud that requests explanations, which in a normal usability testing context could threaten the validity of a user test through reactivity. However, we believe that requesting analysts in usability inspection to produce explanations could improve predicted problem sets [[4](#bib4)]. One plausible outcome is that evaluators make better choices of tactics or strategies for discovering problems, e.g., using usage-focused rather than system-focused approaches [[2](#bib2)]. There are further possible benefits, such as improved learning that benefits future inspections, better use of personal knowledge resources, or lower merging costs that could offset increased observer costs (observers can merge problem instances during/after inspections).

2 Methodology

The methodology was inspired by the falsification testing approached outlined in [[29](#bib29)]. [Figure 1](#fig1) depicts our overall research process. Phase One consisted of the inspections that were elicited either in think-aloud or silent working, with no request to verbalise. The problem predictions arising from this phase were matched and merged to form a master list. The master list of predictions was tested, for accuracy, in a falsification usability test in Phase Two. For simplicity we first describe the inspection phase with its associated results and then describe the falsification test and results.

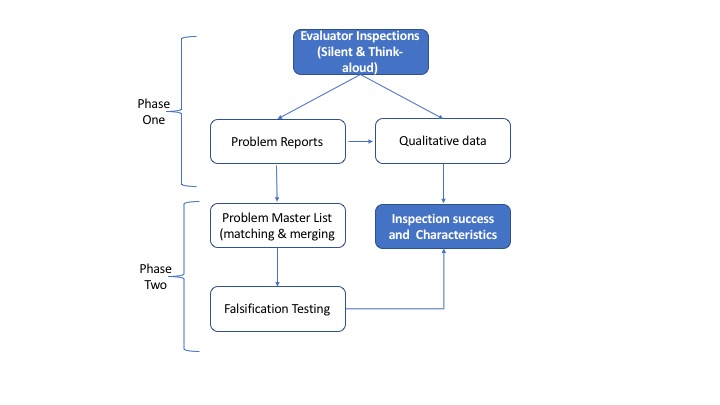


Figure 1: The Research Methodology

2.1 Phase One: Usability Inspection

The usability inspection phase focused on the use of Nielsen’s Heuristic Evaluation [[24](#bib24)]. Here we describe the study methodology and present our analysis of the outcomes of the analysts’ inspections, prior to the falsification phase.

2.1.1 Inspection Participants

Two groups of participants were used in the inspections: analysts and facilitators. Participation was voluntary and no direct incentives were provided to either group.

Analysts: 12 student analysts from a specialist final year degree module in User Experience Design participated in this study. Their ages ranged between 22 and 24 years; 8 males and 4 females. Each had completed 3 previous inspections that were graded as part of course activities. All analysts had studied HCI modules over a three-year undergraduate programme and had completed inspections during each year. Within their final year the students had approximately 8 hours of instruction and practice on usability inspection. We accept that expert evaluators will identify more problems than novice evaluators [[17](#bib17)] however, we contend that our choice of student analysts was justified given the training implications of the study and that a number of highly cited studies in the field are based upon student analysts [[2](#bib2), [6](#bib6),[12](#bib12),[15](#bib15),[31](#bib31)]

Facilitators: 6 students (4 males, 2 females) served as inspection facilitators. They were responsible for issuing instructions and making records. The students had received several practice sessions on test facilitation. The facilitators were drawn from the same undergraduate programme as the analysts. We used simultaneous testing, necessitating the use of multiple facilitators, in order to reduce the possibility of analysts sharing the results or information about the study with each other. Each facilitator observed one analyst in the morning session and a second, different, analyst in an afternoon session. Moreover, we needed to ensure testing (inspection and the subsequent falsification study) could be conducted over a short period of time to reduce the possibility that the test products might be modified by their owners.

2.2 Inspection Design

A repeated measures design was used with the independent variable of inspection type: Silent working (no required verbalization) and Think-aloud (as a concurrent think-aloud with an explicit instruction requesting explanations). The instruction to please-explain, provided in the procedure section below, was specifically designed to invite reactivity: a change in cognitive processing that can result in a change in task performance [[7](#bib7)]. This instruction was used because of recommendations in the literature [[1](#bib1), [28](#bib28)] and the *“please explain …”* prompt, which accompanied the instruction, was used because it is less likely to be subject to facilitator differences: the literature suggests a range of explanation prompts that can be used but skill in their deployment would be necessary and therefore, if used, may introduce a confound [[1](#bib1),[21](#bib21), [28](#bib28)].

Half of the participants inspected aloud first and then worked silently, the other half worked silently and then inspected aloud. The inspection was conducted on two linked websites (www.nexus.org) and (www.traveline.info) which were counterbalanced across conditions, meaning that both sites were inspected in silence and think-aloud. These two, connected sites both provided information on local travel options. We believed these were representative products in that they did not require specific domain knowledge that could be relevant during the inspection. Also, both sites had large user groups to facilitate recruitment of representative users for the falsification study.

The dependent variables all related to the nature of problem predictions made by the analysts. Specifically, we were interested in the impact of thinking aloud on:

 the number and type of problem predictions;

 problem discovery strategy;

 use of knowledge resources used in problem discovery.

The accuracy of the problem predictions would be tested in the falsification study in phase 2 ([Section 2.6](#sec2Z6)).

2.2.1 Problem Reporting

At the end of each individual inspection, our analysts documented their problem predictions using an amended extended structured problem report [[3](#bib3)]. These reports were used to facilitate problem matching and merging (within the falsification study, [Section 2.6](#sec2Z6)). The use of problem reports had been included within the teaching of usability inspection methods, so all analysts were familiar with them. Each report asked evaluators to record:

 A problem description

 Likely user difficulties

 Assumed Causes

 The nature of the problem: Dialogue, Functionality, Navigation or Layout

Additionally, the analysts were asked to indicate how they discovered each problem prediction. Specifically, we used the discovery strategies list below that were identified by [[2](#bib2)]. We were open to the possibility that other strategies might be used and therefore we included an “other” category that analysts could complete if they believed their strategy was not represented.

 *System Scanning*: looking around the system in no particular order

 *System Searching*: Sequentially examining various toolbars menu options and form elements

 *Goal Playing*: Trying to achieve a user goal using the affordances on the website

 *Method Following*: Achieving a goal according to a method determined by analyst expectations rather than website options.

2.3 Inspection Procedure

The inspections were conducted over the period of one day. The 12 analysts were divided into two groups, with each analyst performing two separate inspections. The first group of six conducted their two inspections during the morning and the second group of six conducted their two inspections during the afternoon. Each individual inspection lasted around 45 mins meaning that the analysts worked for approximately 1.5 hours. The analysts were seated separately in different teaching cells so that they were unable to hear the others’ verbalizations or to see the inspection activities that other analysts were performing. This process necessitated the use of facilitators whose role was to issue instructions; prompt for explanations (depending on the study condition) and to make contemporaneous notes. The same facilitators were used in both conditions (silent and think-aloud and in both morning and afternoon sessions) for purposes of standardization.

Following the signing of the necessarily consents, the analyst participants were informed that we were interested in learning more about how the process of usability inspection and the type of issues that are typically highlighted. The facilitators used cue cards to ensure the instructions were the same. The analyst participants were not informed that the focus of the study was the impact of think-aloud on inspection until the end of the second inspection.

For the silent working condition, analyst participants were shown the test website and were instructed to inspect the product using heuristic evaluation, which they had been taught in class and had practiced previously in tutorials. To remind them of the heuristics they were given a printed list that described each of the ten heuristics. The facilitator sat a little way behind and to the right of the analyst.

For the think-aloud inspection, analyst participants were shown the test website and were instructed to inspect the product and explain aloud their thinking. The instruction used was *“Please explain your thinking as you inspect the product; tell us what you are doing and why; explain any potential problems you find”* The observer would prompt the analyst to *“please explain”* if they fell silent for 15 seconds.

In both conditions the analysts were asked to inspect the website for 45 minutes. Regardless of condition, analysts were allowed to make notes on blank paper if they felt necessary, and were given a collection of blank problem reports to record their problem predictions. We did not set a target number of reports to be completed so as to not create undue pressure.

Analysts were asked to review their reports and give severity ratings. This included 0 to indicate that a *possible* problem should be eliminated, leaving only *probable* problems. Finally, each analyst was asked about which inspection approach they preferred and the reasons for their preference. Participants were debriefed at the end of the study in a classroom setting.

2.4 Phase One Inspection Results

First, we describe the nature and type of problem predictions that were made and the reported discovery methods used. This analysis is based upon all the predictions made by each individual analyst prior to duplicates being removed through matching and merging. The analysis does not consider the accuracy of the problem predictions, which is considered in the analysis of the falsification study reported in [Section 2.6](#sec2Z6).

2.4.1 Number of predictions made

Overall, including duplicates, 62 initial problem predictions were produced. A Wilcoxon signed ranks test revealed a significant difference between the two inspection conditions (Z = -2.57, p < 0.01). Significantly more possible problems were initially identified when analysts engaged in normal silent working (mean = 3.17; sum = 38) than during the think-aloud condition (mean = 2.0; sum 24).

2.4.2 Reported Discovery Method

[Figure 2](#fig2) shows the proportion of predictions that were made using each discovery method by inspection condition. Overall, the analysts used a range of problem discovery methods with 15 (24%) predictions arising from system scanning; 17 (27%) from system searching; 21 (34%) from goal playing and 9 (15%) from method following. Proportionally, the use of system scanning was similar between the conditions, but system searching was greater in the silent condition and goal playing was the most prevalent strategy in the think-aloud condition.

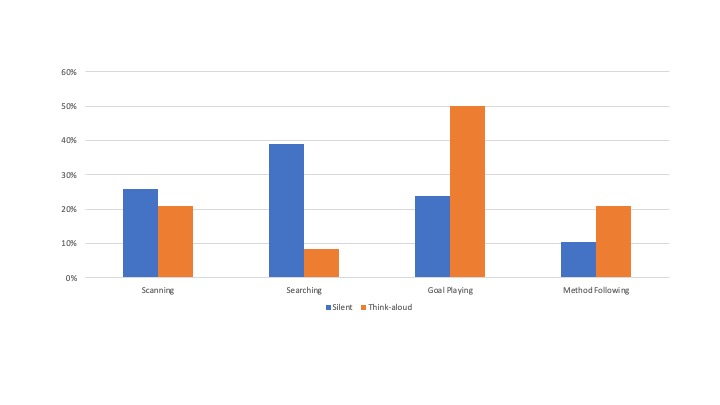


Figure 2: Percentage of reported discovery methods used in the silent and think-aloud condition

2.4.3 Reported Problem Types

[Figure 3](#fig3) presents problem types by inspection condition. Overall, there were: 26 Layout; 16 Functionality, 11 Navigation and 9 Dialogue. Proportionally, the think-aloud condition produced more Functionality, Layout and Dialogue problems that the silent condition. However, a greater proportion of Navigation problems arose from the silent condition.

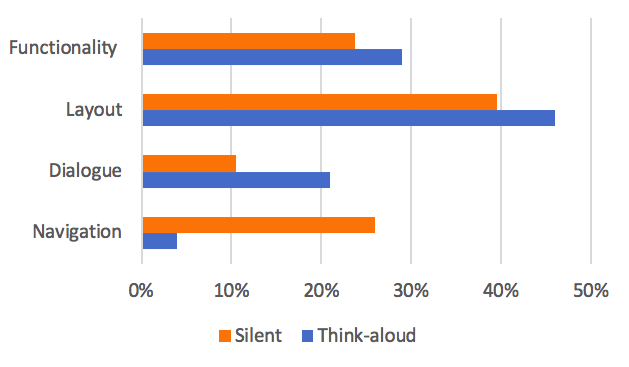


Figure 3: Percentage of reported problem types in the silent and think-aloud condition

[Figure 4](#fig4) presents the number of each type of problem prediction by the discovery method used. Overall, there seemed to be a more prominent method for each type of problem: scanning led to more layout predictions; searching was most prominent in navigation predictions; goal playing for dialogue and method following for functionality.

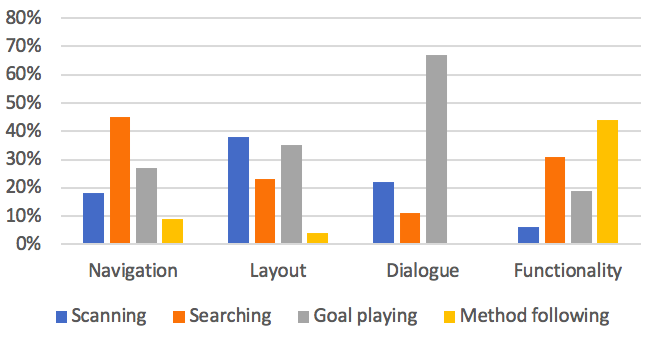


Figure 4: Percentage of reported problem types predicted in the silent and think-aloud condition by discovery strategy

2.4.4 Analyst Preferences

The analysts marginally (n=7) preferred the think-aloud inspection over the silent working inspection (n=5). Reasons for preferring silent working were twofold: two analysts said that it felt artificial or unnatural to think-aloud while they worked, and three analysts cited concerns that their competency (as determined by what they said) might be judged by the facilitators. For example, *“I didn’t want to appear stupid, you know if my problems weren’t great, so I preferred just working quietly”.* Those analysts who preferred think-aloud highlighted that it helped them to clarify their thinking (n=3), helped them to complete the problem reports (n=1) and made them more active (n=3). However, this activity may have arisen, in part, as an artefact of the social dynamic. For example, one participant commented *“I felt like I needed to really engage so I started to try and achieve more in the way of tasks rather than just plough through links”.* One participant added a caveat to her preferences: “*explaining to someone else* *helped but they could have noted the problems and I might have found more in the time we had*”. This analyst wanted the listener to act as a scribe as she (in a later comment) indicated that explaining the problem then writing it down duplicated effort that she could have spent finding more possible problems.

2.5 Discussion of initial inspection results

Overall, our analysts made fewer problem predictions than participants in other studies such as [[29](#bib29), [31](#bib31)]. However, these studies used more participants who worked in groups, for longer time periods, and they do not report a mean discovery rate per analyst, so we cannot be sure as to what the individual discovery rate would have been had the participants in these other studies worked alone.

The silent condition produced significantly more initial predictions than the think-aloud condition. This finding did not surprise, as studies comparing concurrent think-aloud within usability testing and silent working [[20](#bib20)] often report extended task completion times for the think-aloud condition, simply because thinking aloud slows users down and therefore we might expect, in the current context, that would have the corollary of reducing prediction yield.

The reported discovery mechanisms suggest that the act of thinking-aloud was associated with different problem discovery strategies. The most used strategy in the silent condition was system searching, where an analyst will systematically try out interface features. The most used strategy in the think-aloud condition was goal playing where analysts try to accomplish user tasks. In common with previous studies [[31](#bib31)] scanning featured prominently in both conditions. The increased use of goal playing in the think-aloud condition is interesting in that previous studies reported lower figures for this discovery method [[2](#bib2)]. The think-aloud condition seems to have made analysts think more about users and their goals; activating a resource that is outside of the inspection “method”. Method following was the least used strategy in both conditions, but the numbers were higher than [[31](#bib31)]. Again, we believe this may be because of explanations within the think-aloud, given the proportionally greater use of this strategy was reported in this condition. However, there is an alternative explanation. We believed the original definition of method following was unclear and we amended this in our problem reports to clearly delineate it from goal playing as being expectation led. This enhancement to the definition of method following may have helped analysts to be clearer in determining which of the two more active discovery strategies they had used. However, we also have to accept that asking analysts to indicate their discovery method (as other studies have done [[30](#bib30)]) may, in itself, have encouraged analysts to change their discovery method, since listing the methods might have cued performance. However, we believed it was better to use analyst self-report on discovery method rather than ask facilitator to code for this, as the use of multiple facilitators might have introduced inconsistencies in coding.

The analysts reported finding a range of problem types. Proportionally, the pattern that emerged was similar for both conditions, although the think-aloud condition generated relatively fewer navigation problem predictions. This may have occurred because system searching, the most reported strategy in the silent condition, involves systematically trying links and therefore might be the reason for the differences.

2.5.1 Preferences

Analysts expressed a slight preference for inspecting while thinking-aloud (7 vs. 5). The preferences for think-aloud support the view that self-generated explanations may help to fill gaps in thinking [[1](#bib1),[20](#bib20)]. A content analysis of the explanations is needed to confirm this in a follow-up study. However, the reasons for preferring the silent working condition were slightly more worrisome with regard to social perceptions of analyst competency. Laying bare one’s thinking might invite judgements from the listener, which may have the impact of stifling the creative process or filtering and editing of verbal protocols: issues that are also believed to effect think-aloud verbal reports [[5](#bib5)]. The impression management concerns of our analysts may be reflective of their relative youth and stage of HCI expertise and may not appear in a professional context. However, this finding is worthy of further investigation in terms of the construction of the inspector-listener pairs, for example factoring gender, past experience or by modifying how the dyads work together, for example using constructive interaction scenarios [[27](#bib27)].

2.6 Falsification Study Methodology (Phase Two)

A falsification study was designed to stress the predictions made in phase one and support coding in line with the 5 logical outcomes of inspection [[31](#bib31)]. In order to test the predictions made during the inspections it was necessary to form a master list of unique problem predictions that could be tested for their accuracy within a usability (falsification) test. We describe this process before presenting the falsification test procedure and results.

2.6.1 Master Problem Set Development

Problem merging is important for eliminating over reporting by preventing a single usability problem, with multiple instances, being erroneously classed as two or more different ones. The primary tool used to support problem matching was the extended structured problem report, which has been shown to effectively support merging [[14](#bib14)]. The first author merged problem predictions by comparing their descriptions, the context in which they arose, their identified likely causes and mentioned features. The process was conducted in three stages and was conducted over a period of two weeks.

In the first stage, the first author carefully read each problem report one-by-one, when she encountered a report that was similar to a previously read one, it was grouped with it for later analysis. This resulted in several groups of similar reports, plus the remaining single reports.

In the second stage, focusing on the groups of similar reports the first author read each report again one-by-one and carefully matched reports based upon their descriptions, context and causes. Using an excel spreadsheet she constructed a table that included a short title for each problem and then recorded the problem IDs next to the title.

In the third stage, two weeks later the entire set of reports was re-read again using steps one and two above. This process of problem merging reduced the set of 62 problem reports to 43 unique problem predictions. While the first author was responsible for organising the inspection sessions she was not involved in facilitating the inspections and, therefore, was not influenced by events from the sessions.

2.6.2 Task construction

Using the problem titles as an initial point of guidance and re-reviewing problem reports as needed, the first author constructed a task set that was designed to lead users to the areas of the sites that the analysts predicted would be problematic. In some instances, the tasks were derived from the context statements in the problem reports (the situations or activities where the analyst believed the problems would arise). Falsification testing stresses features associated with predicted problems several times across a task set and is therefore designed to ensure comprehensive coverage. A single task may lead users to more than one problem; therefore, the number of tasks does not equal the number of predictions made. For example, the following task tested predictions 4, 29, 30, 31, 33, 34 listed in [Table 1](#tb1) section 2.7.1.

“You are organising a day trip for approximately 10 children, all under 16 years old. You will need to travel by metro. What type of ticket would be most economical?”

In all 25 tasks were constructed, some were multi-part tasks. These tasks mapped to 41 of the 43 unique problem predictions. We did not construct tasks to lead users to 2 of the identified predictions (Predictions 9 and 11 in [table 1](#tb1) section 2.7.1) as they related to broken functionality (e.g. broken hyperlink). The falsification study had to focus on those predictions that were testable conjectures. Neither of these problems could be falsified by any user test, so no task factored them in.

2.6.3 Falsification Test participants

In total, 12 volunteers, 6 males and 6 females, aged between 18 and 68 participated. All were representative users of the test products as determined by a user profile questionnaire. Participants were recruited from the local population. Each person received a £10 shopping voucher to thank them for their time and travel expenses. We believe this level of incentive is unlikely to affect the results.

2.6.4 Falsification Test procedure

A Research Assistant (RA) welcomed each participant to the usability laboratory and the purpose of the study was explained to them as a usability test of the study websites. Participants were asked to complete all 25 tasks. The test websites were linked products with their own individual tasks. Half of the participants started with Site A tasks and then Site B tasks and vice versa.

The participants were asked to work on each task sequentially and unaided. Tasks were handed to the participants on printed cards by a researcher who sat in the room with them. Participants were asked to think-aloud using Ericsson and Simon’s Classic method [[5](#bib5)]. No prompts other than “please keep talking” were issued. After each completed task, participants were asked to rate the task difficulty on a 5-point scale ranging from easy to very difficult. The participants’ screen and voice were recorded using TechSmith Morae.

2.6.5 Falsification Data Analysis

The test sessions were reviewed for evidence to support/refute the analyst predictions. We used the following indicators of users experiencing difficulties

 Verbal Evidence: a verbal statement from the user indicating a problem

 Behavioral Evidence: e.g: abandonment; missed links or a feature they were supposed to interact with; long task completion times

 Subjective assessment from task difficulty ratings

The RA reviewed each recording to identify evidence of user difficulties and noted the evidence, where present, on a problem report. [Figure 5](#fig5) shows an example individual report for one test user and one task

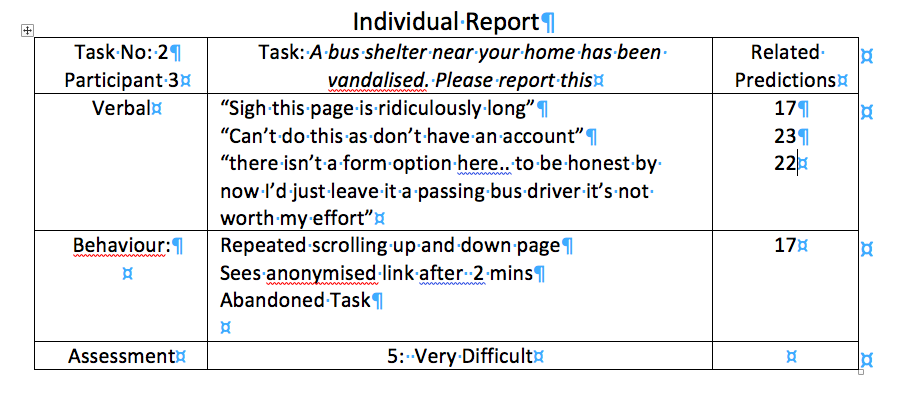


Figure 5: Example falsification test problem report showing instances of verbal evidence, behavioural evidence and user difficulty assessment

To foster consistency and manage the Evaluator Effect we used a breakdown checklist [[17](#bib17)] which specified examples of usability break downs. The RA and first author coded the first test user together, the RA then independently coded the test sessions. Once all the sessions had been reviewed, the RA collated evidence by task across all 12 users. The recordings for each test user were carefully reviewed and breakdown evidence was noted on a problem report on a task-by-task basis (e.g. [Figure 5](#fig5)). To give an indication of reliability, the first author also independently coded the data of 2 participants and the any-two agreement [[9](#bib9)] figures were (68% and 62%). This is comparable to other published works [[6](#bib6), [21](#bib21), [32](#bib32)]. The first author then reviewed the reports against analyst predictions. Taking each prediction and associated tasks she recorded where evidence supported the prediction. For a prediction to be supported two sources of evidence from the list above were required, with this evidence relating to the specific feature mentioned in the analysts’ problem report descriptions. Where breakdown evidence was found that did not concur with any analyst prediction then a true miss was recorded. The coding of true misses was conducted after predictions had been noted on the problem reports. The RA cross-checked this process.

Finally, the reports were coded for the use of knowledge resources. We used the following list of 6 knowledge resources: User, Task, Technical, Domain, HCI, and Design [[31](#bib31)] as our coding scheme. The first author and the RA coded the problem reports together in a joint session. Highlighting the relevant extracts in each report and recording the knowledge resource used against each prediction in an excel spreadsheet.

2.7 Falsification Study Results

We first describe the characteristics of the Master Set of problems and then consider the interplay between the reported discovery method, use of knowledge resources and the accuracy of the predictions.

2.7.1 Master Set Characteristics

[Table 1](#tb1) presents the master set predictions that were unique to the silent condition. [Table 2](#tb2) presents the master set predictions that were unique to the think-aloud condition. [Table 3](#tb3) presents the predictions that were made by both conditions; these are subsequently referred to as overlapping predictions. Additionally, 14 problems were predicted by multiple analysts; these are indicated with an asterisk in [Tables 1](#tb1), [2](#tb2) and [3.](#tb3)

Tables 1-3 also present the results of the falsification test. For each prediction ID, the tables present a short summary of the problem prediction; whether analysts confirmed or eliminated the problem; and the falsification outcome [[29](#bib29)].

Overall, a process of problem merging reduced the set of 62 problem reports to 43 unique problem predictions. Within that set of predictions, 22 were unique to the silent working condition. However, 2 of the predictions (9 and 11) made in the silent condition (last two rows of [Table 1](#tb1)) were not subject to falsification testing as they related to broken functionality (e.g. broken hyperlink). The think-aloud condition yielded 15 unique predictions. Finally, 6 predictions were made in both conditions. Therefore, removing problems (9 and 11) a total of 41 predictions were subject to falsification testing.

With analyst elimination of improbable problems, falsification results could be coded as follows [[2](#bib2)]

 *True Positive*: The analyst confirmed the problem in the review session at the end of the inspection and evidence was found to support the prediction within the usability test.

 *True Negative*: The analyst eliminated the problem in the review session and no evidence was found to support the prediction in the usability test.

 *False Positive*: The analyst confirmed the problem in the review session and no evidence was found to support the prediction in the usability test.

 *False Negative*: The analyst eliminated the problem in the review session, but evidence was found to support the prediction in the usability test.

 *True Miss*, a problem revealed in testing that was not predicted during the inspection process.

Tables 1-3 pertain only to the predictions made and do not contain True Miss problems where a problem arose in falsification testing that was not predicted. In total, 4 true miss problems were identified from testing. These problems were:

 Unnecessary confirmatory step in journey destination selection (3 falsification test participants)

 Text colour providing poor legibility (2 falsification test participants)

 Insufficient explanation on transport terms (1 falsification test participants)

 Poor visibility of anonymous reporting (5 falsification test participants)

Table 1: Master Set problems that were unique to the Silent Condition

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Prediction summary | Analyst  Decision | Falsification  Result |
| 3 | Menus include options not relevant to primary users | Confirmed | True Positive |
| 5\* | Information for travel support is at deep levels in hierarchy | Confirmed | True Positive |
| 12\* | The timetable layout is confusing | Confirmed | True Positive |
| 20\* | Pages do not facilitate scanning and require significant reading | Confirmed | True Positive |
| 31 | Tickets are misclassified undermining search results | Confirmed | True Positive |
| 32\* | Ticket zone information is buried | Confirmed | True Positive |
| 37 | Quick journey planner label is too similar to journey planner | Confirmed | True Positive |
| 14\* | The boundaries between linked sites are unclear | Eliminated | True Negative |
| 15 | There is no exit; user must rely upon back button | Eliminated | True Negative |
| 16 | Horizontal scrolling on map | Eliminated | True Negative |
| 24 | Feedback link confusing is it service or website | Eliminated | True Negative |
| 42 | Route information is limited | Eliminated | True Negative |
| 43 | Journey planner icons are unclear | Eliminated | True Negative |
| 1 | Lower menu link labels have poor information scent | Confirmed | False Positive |
| 2 | The site does not utilize breadcrumbs to aid navigation | Confirmed | False Positive |
| 4 | There is no FAQ or help page | Confirmed | False Positive |
| 6 | Support cards are referred to using different terminology | Confirmed | False Positive |
| 8 | Images can be confused as links | Confirmed | False Positive |
| 30\* | Ticket finder has different designs depending on access point | Confirmed | False Positive |
| 27 | Ticket finder defaults not matched to common options | Eliminated | False Negative |
| *9* | *The link to the pop card information is broken* | Confirmed | *Not Tested* |
| *11* | *The search facility does not work* | Confirmed | *Not Tested* |

Table 2: Master Set problems that were unique to the Think-aloud Condition

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Prediction summary | Analyst  Decision | Falsification  Result |
| 7 | Menu organisation does not reflect user priorities | Confirmed | True Positive |
| 18 | Disruption box dominates page and is blank | Confirmed | True Positive |
| 19\* | Downloadable links are obscured below the fold | Confirmed | True Positive |
| 21 | Transport submenu lack visibility | Confirmed | True Positive |
| 22 | Incomplete section labels for issue reporting | Confirmed | True Positive |
| 23 | Contact signing via login will be barrier to use | Confirmed | True Positive |
| 34 | Auto reset on ticket finder does not support error correction | Confirmed | True Positive |
| 38 | Map based navigation relies too heavily on local knowledge | Confirmed | True Positive |
| 39 | Static map infers geolocation functionality that does not exist | Confirmed | True Positive |
| 41\* | Myjourney field suggest map use | Confirmed | True Positive |
| 35 | Journey planner form includes inconsistencies | Eliminated | True Negative |
| 40 | Myjourney map symbols are not clear | Eliminated | True Negative |
| 33 | Ticket finder returns lack organisation | Confirmed | False Positive |
| 36 | Myjourney does not adequately describe tool functionality | Confirmed | False Positive |
| 26 | Journey details box font size is too small | Eliminated | False Negative |

Table 3: Master Set problems that were in both Conditions (Overlapping)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Prediction summary | Analyst  Decision | Falsification  Result |
| 10\* | The home page contains too much information | Confirmed | True Positive |
| 13\* | Metro maps not placed at the point of need | Confirmed | True Positive |
| 17\* | Excessively long pages | Confirmed | True Positive |
| 25\* | Section menu text looks like a heading rather than a link | Confirmed | True Positive |
| 28\* | Icons do not match selected filters on ticket finder returns | Confirmed | True Positive |
| 29\* | Filter options do not represent full range of tickets | Confirmed in Silent | False Positive |
| 29\* | Filter options do not represent full range of tickets | Eliminated in Think-Aloud | True Negative |

Problem 29 was retained by one analyst, but eliminated by another

2.7.2 Predictions that were unique to each condition

As noted above, 22 predictions were unique to the silent condition and 15 were unique to think-aloud, with 6 ‘overlap’ predictions made by both. [Table 4](#tb4) shows the number of confirmed analyst predictions and eliminations. [Table 5](#tb5) summarizes the falsification results. Within both [Table 4](#tb4) and 5 the results are paired in rows. The upper row of each pair is for the predictions that were unique to a condition. The lower row includes the overlap predictions that were made in conditions. Problem 29 has been omitted from this analysis and is discussed in the next section. Therefore, the number of overlap problems used is 5 rather than 6. For all other problems predicted by multiple analysts there was agreement in the reports as to whether to retain or eliminate the problem.

Table 4: Analyst decisions for the unique and overlap predictions

|  |  |  |
| --- | --- | --- |
| Inspection Condition | No. of retained predictions | No. of Eliminations |
| Silent (unique, n=20) | 13 (65%) | 7 (35%) |
| Silent with 5 Overlap Predictions (n=25) | 18 (72%) | 7 (28%) |
| Think-aloud (unique n=15) | 12 (80%) | 3 (20%) |
| Think-aloud with 5 Overlap Predictions (n=20) | 17 (85%) | 3 (15%) |

Excludes predictions 9,11 and 29; total number of predictions = 40

Table 5: Falsification Results for the unique and overlap predictions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inspection Condition | True Positives (TP) | False positives (FP) | False negatives (FN) | True Negatives (TN) |
| Silent (unique, n=20) | 7 (54%) | 6 (46%) | 1 (14%) | 6 (86%) |
| Silent with 5 Overlap Predictions (n=25) | 12 (67%) | 6 (33%) | 1 (14%) | 6 (86%) |
| Think-aloud (unique n=15) | 10 (83%) | 2 (17%) | 1 (33%) | 2 (67%) |
| Think-aloud with 5 Overlap Predictions (n=20) | 15 (88%) | 2 (12%) | 1 (33%) | 2 (67%) |

Excludes predictions 9,11 and 29; total number of predictions = 40

Overall, a greater proportion of problems were confirmed as correct by analysts when working in the think-aloud condition with, proportionally, fewer eliminations made than during the silent condition. The falsification results suggest that proportionally the TA condition made more accurate predictions and had fewer false positives. The silent condition, however, had a greater proportion of correct eliminations. This may reflect the poor quality of the initial predictions made during discovery.

2.7.3 Multiple predictions and overlap problems

Of the predictions made by multiple analysts 79% (n=11) were TP; 7% (n=1) were FP; 7% (n=1) were TN. Problem 29 had conflicting analyst decisions (retained in silent condition as a FP and correctly eliminated in the think aloud as TN). The other 5 problems that were found in both inspection conditions were confirmed by analysts and were coded as True Positives following the falsification study.

2.7.4 Reported discovery strategy of successful predictions

[Table 6](#tb6) shows the frequency of each discovery method used for those predictions that were successful (True Positives and True Negatives). As some predictions were made by more than one analyst the total number of discovery method frequency is greater than the number of predicted problems. Overall, goal playing dominated the think-aloud condition. The least used strategy for successful predictions, in this condition, was system searching. For the silent condition, scanning and goal playing were the most used discovery methods.

Table 6: Discovery Method used in successful predictions

|  |  |  |
| --- | --- | --- |
| Discovery Method | Silent Inspection | Think-Aloud Inspection |
| Scanning | 7 (35%) | 2 (11%) |
| Searching | 4 (20%) | 1 (5%) |
| Goal Playing | 6 (30%) | 10 (55%) |
| Method Following | 3 (15%) | 5 (28%) |

Some predictions were made by more than one analyst

2.7.5 Knowledge Resources

[Tables 7](#tb7) and [8](#tb8) show the instances of correctly and incorrectly applied knowledge resources for the “true positive” predictions for the silent and think-aloud inspection condition. All analyst predictions were coded for knowledge resource [[14](#bib14)] use (including duplicates, Problems 9 and 11 excepted as above). Hence the number of predictions used in this analysis is 60 and not 62. Overall, analysts correctly used 5 individual types of knowledge resource: user, domain, HCI, Design and Task. There was no evidence of individual use of technical knowledge, although it was successfully combined with user knowledge by one analyst. Given the low number of true predictions, the instances of each individual knowledge resource are correspondingly low.

Table 7: Successful application of individual KRs on True Positive Predictions

|  |  |  |
| --- | --- | --- |
| Knowledge Resource | Silent Inspection | Think-aloud Inspection |
| User | 0 | 1(5%) |
| Task | 1(5%) | 0 |
| Domain | 2(10%) | 0 |
| HCI | 6(30%) | 3(16%) |
| Design | 10(50%) | 4(22%) |
| Technical | 0 | 0 |

Table 8: Unsuccessful application of Individual KRs on True Positive Predictions

|  |  |  |
| --- | --- | --- |
| Knowledge Resource | Silent Inspection | Think-aloud Inspection |
| User | 0 | 3(50%) |
| Task | 0 | 0 |
| Domain | 3(19%) | 0 |
| HCI | 7(44%) | 2(33%) |
| Design | 5(31%) | 1(16%) |
| Technical | 0 | 0 |

However, as [Table 9](#tb9) shows technical knowledge was successfully combined with user knowledge by one analyst. Given the low number of true predictions, the instances of each individual knowledge resource are correspondingly low, however there was more evidence of a spread of knowledge resource use in the Think-aloud condition with 53% (n=10) of correct predictions combining two knowledge resources compared to only 5% (n=1) in the silent condition.

Table 9: Successful use of joint KRs on True Positive Predictions

|  |  |  |
| --- | --- | --- |
| Knowledge Resource | Silent Inspection | Think-aloud Inspection |
| User & Design | 0 | 2(11%) |
| User & HCI | 0 | 4(22%) |
| User &Technical | 0 | 1(5%) |
| Domain & User | 1(5%) | 1(5%) |
| HCI & Design | 0 | 0 |
| Design & Domain | 0 | 1(5%) |
| Task and HCI | 0 | 1(5%) |

3 Discussion of Falsification outcomes

Overall, while the think-aloud condition made fewer predictions it had more confirmed predictions in the falsification study. Moreover, the successful predictions (true positives) in the think-aloud condition showed evidence of analysts using more than one type of knowledge resource. This suggests that the explanations generated during the think-aloud were associated with the activation of different knowledge types [[1](#bib1), [21](#bib21)]. Interestingly, the silent condition generated more correct eliminations than the think-aloud condition (35% vs 20%). This could suggest that during a think-aloud, analysts were applying their knowledge resources more rigorously as they were more convinced by the (more accurate) issues they reported, and so may have had less need to eliminate in comparison to the silent condition. Some of the issues that were correctly eliminated in the silent condition, the true negatives, could be considered surface problems, that were easier to spot and related strongly to heuristic principles (e.g. no emergency exit; consistency)

The silent inspection also had proportionally more false positives than the think-aloud inspection. Some of these could have been created due to a lack of reflection on the retained predictions. For example, in Problem 6, “Support cards are referred to using different terminology”. The analyst incorrectly believed that two different travel cards were one and the same thing showing a misunderstanding of the domain. The information pertaining to the cards was distributed within a large block of text which did not facilitate scanning (Problems 17 and 20) which possibly lead the analyst to extract incorrect information about the cards. The test users within the falsification study however, while complaining vociferously about the amount of reading and page length, did manage to differentiate between cards. Therefore, the analyst may have reported a specific example of a user difficulty (misunderstanding the text) arising from Problems 17 and 20 as a problem in its own right.

The falsification testing also enabled us to examine the discovery strategies with successful predictions (True Positives). As noted previously, during initial discovery of *possible* problems the silent condition favoured system searching but this was not associated with relatively more accurate predictions. Scanning, closely followed by goal playing was the most successful strategy used. Scanning was likely to be successful as a predictor of layout-based issues (e.g. problems such as 10, 12,). In the think-aloud condition Goal Playing was the most successful strategy and the use of Method Following also led to an increase in the number of successful predictions. Effective use of scanning in the silent condition indicates that the discovery methods do have effective roles to play. However, goal playing and method following do seem to encourage analysts to take a more user-centred focus and utilize their own knowledge resources to better effect.

4 General Discussion

The present study explored the impact of asking analysts to think-aloud during usability inspection on: number and type of problems they predict; discovery method, use of knowledge resources, and accuracy of predictions made. Overall, our findings suggest that while fewer initial problems predictions were made in the think-aloud condition: there were fewer eliminations; problem discovery methods were more user-centred; a greater spread of knowledge resources were used; and predictions were, proportionally, more accurate. We now discuss results and their implications in more detail.

We used an instruction that would request explanations because we wanted to highlight the activation of higher order thinking rather than task-based accounts of what the analyst was actually doing. It would have been possible to have analysts self-explain to themselves but we believe the presence of a listener is an effective aid to both maintain the verbalisation and also to listen and note issues as they arise. Indeed, some of the qualitative comments suggested the presence of listener had an impact on the analysts’ activities.

Overall, the silent condition produced more problems than the think-aloud condition (38 vs 24 respectively). However, problem yield is not the only measure of method effectiveness [[11](#bib11)]. In terms of accuracy, the falsification test of the merged problems indicated that the think-aloud condition produced a greater proportion of correctly retained predictions (83% vs 54%) and a small number of false positives (17% vs 46%). In terms of validity (number of real problems found by the method divided by number of predicted usability problems) the silent condition (including overlap predictions) results suggest a score of 0.66 (12/18), and think-aloud a score of 0.88 (15/18). In terms of thoroughness (number of real problems found divided by number of real problems that exist, which includes true misses and false negatives), the silent condition suggests a score of 0.42 (12/28) and the think-aloud a score of 0.54 (15/28). The overall effectiveness of each (thoroughness x validity) is silent 0.28 and think-aloud 0.48. Overall, the scores are higher for the think-aloud condition, although they are not as high as some studies [[31](#bib31)]. However, this may be due to differences in procedures. We used two types of inspection: silent vs. think-loud with analysts working alone for a limited window of time. Other, notable, studies such as [[31](#bib31)] used larger numbers of novice evaluators working in groups on a single inspection for a longer period of time.

Moreover, the discovery method analysis suggested that the think-aloud approach increased the use of more user-centred and active discovery strategies. The think-aloud condition produced fewer eliminations overall, suggesting that the analysts were more convinced that their predictions were probable (and more credibly so), suggesting a more active strategy. Indeed, the reported discovery method for each confirmed prediction suggests that think-aloud promoted more active discovery where analysts were engaged in trying to accomplish user tasks.

So, what could account for the differences in performance? Within the instructional design literature there are several alternative explanations as to how the generation of explanation works. VanLehn et al. [[28](#bib28)] suggest that gap filling is the most probable explanation: learners (in our case analysts) actively fill gaps in their own knowledge by making better use of pre-existing knowledge about users, tasks and the domain. In other words, explanation encourages them to better exploit their personal knowledge in making sense of the system. However, we must also consider an alternative prospect. The analysis of the analyst preferences might suggest that the presence of the facilitator promotes a Hawthorne effect. Analysts know they are being heard and their behaviour changes accordingly and they become more active. However, this is still a useful, positive, effect that could have implications for how we teach inspection, perhaps through the use of teach-back scenarios or constructive interaction [[27](#bib27)]. Indeed, previous studies in the field [[29](#bib29), [31](#bib31)] have not monitored analysts; they have simply analysed the fruits of their labour: problem reports. Finally, the explanation may be based on the point at which explanations operate during the inspection process. In his seminal work on reflective processes, Schön [[26](#bib26)] distinguishes between reflection *in* action and reflection *on* action. The generation of explanations and the subsequent writing of problem reports might activate both processes whereas the methodologies of previous studies may have only activated reflection *on* action.

Schön’s research [[26](#bib26)] concentrated on creative professionals. The primary beneficiaries of our work would be students of HCI or novice inspectors seeking to improve their practice. Future studies could examine scaffolded support for explanation generation through targeted probes from educators that might seek for example, to activate particular knowledge resources, such as asking analysts to reflect on how an issue might impact particular user groups. Other possible beneficiaries of our research are not only UX professionals and front-end developers (who apply established usability inspection approaches), but also interaction designers who could use explanation in their studio-based practices as a complement to silent individual reflection. Think-aloud may also have benefits for other inspections, e.g., for accessibility and platform standards and style guide compliance. Of course, conducting inspections in using inspector-listener pairs would add to the cost of what is supposed to be a discount method. However, cost savings might be achieved in the overall time taken to complete the inspection should the listener also act as a problem scribe. Alternatively, it may be fruitful to use an automatic speech-to-text tool to record the think-aloud. This would have the benefit of reducing the number of analysts needed and would overcome a second notable issue; the time taken to complete problem reports [[15](#bib15)]. Further possibilities for software support include problem report databases and associated support tools for problem merging, planning falsification testing, and processing the results of such testing.

4.1 Limitations

There are several limitations with the study. First, we used student evaluators rather than experienced practitioners and this may account for the reason why the problem discovery rate was low. However, several influential studies have been based on student evaluators [see for example, 2, 6,12,15,31]. Moreover, the students had previously conducted several inspections within their final year course and in the previous three years. For us this, coupled with the implications the work has for training, justifies the use of students in this instance. Second, we used only 12 evaluators, which we acknowledge this is a small sample. However, because the experimental manipulation required an observer to be present, practical limitations were imposed on how many inspections we could conduct in a single day. Other inspection studies have used larger samples [[15](#bib15), [31](#bib31)] although some did leverage students working in groups. Ours was a repeated measures study and the sample size is comparable to some influential studies within the TA literature [[8](#bib8)]. Third, while two sites were used, they were linked and pertained to the same topic: supporting local travel. It would have been better to have used sites on different topics. However, we did counterbalance them across conditions to reduce transfer. Fourth, the researcher who administered the falsification study also coded the user test data, so she could have been influenced by the activities in the session. We endeavored to reduce this through the inclusion of a 3-week break between the tests and data analysis but it would have been better to use separate RAs for each activity. Finally, we do not present an analysis of the content of the think aloud in terms of an utterance analysis as used in several think-aloud studies [[21](#bib21), [32](#bib32)]. We are currently conducting a protocol analysis of expert inspectors to examine the nature of the utterances produced and their relationship to problem discovery and analysis.

4.2 Conclusions

We suggest that inspection is a form of learning in which analysts make a deliberate attempt to understand the interface and its domain using their pre-existing knowledge to identify problem predictions and then re-evaluate their own ideas in light of the new material during analysis. As such, we posited that strategies that improve learning, such as providing explanations might also improve inspection. Our study produced some promising results and points to areas of further investigation which are currently underway. However, our results are relevant to education and practice, with immediate potential benefits of using think-aloud during usability and other inspections.

Further investigations are needed to allow more confident recommendations for practice, but it is safe to recommend some use of think aloud in usability inspection. Exclusive use of think-aloud could be an unwise risk, given than 13 of the 30 retained predicted problems were unique to the silent condition. Equally, 12 of the 30 retained predicted problems were unique to the think-aloud condition, and it would also be unwise to lose these. Currently, practitioners and educators can experiment with different mixes of think aloud and silent inspection, perhaps pairing analysts were resources allow and having them merge their problem sets. Another option is for analysts to repeat an inspection, with either silent or think-aloud first. The ability to generate these and other options is evidence of the promise of combining think-aloud with some usability inspection. Our next steps will be to investigate ways to effectively combine inspection with and without think-alouds.

Acknowledgements

We thank members of the review committee for their helpful comments on our manuscript. Thanks, are also due to Shell Young and Alan Lumsden for their help with our research and to the students who served as analysts and facilitators.

REFERENCES

[1] Chi, M.T.H., de Leeuw, N., Chiu, M.H. and La Vancher, C. (1994) Eliciting self-explanations improves understanding. Cognitive Science, 18, 439-477

[2] Cockton, G, Woolrych, A, Hall, L, Hindmarch, M. (2003): Changing Analysts' Tunes: The Surprising Impact of a New Instrument for Usability Inspect. In: Proceedings of the HCI03 Conference on People and Computers XVII , 2003, .pp. 145-162

[3] Cockton, G, Woolrych, A and Hindmarch, M (2004) Reconditioned merchandise: extended structured report formats in usability inspection.In:Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04.Association for Computing Machinery, New York, pp. 1433-1436.ISBN 1-58113-703-6

[4] Cockton, G, Woolrych, A, Hornbæk, Kand Frøkjær, E (2012) Inspection-based evaluations. In: The Human-Computer Interaction Handbook: fundamentals, evolving technologies, and emerging applications [3rd edition]. CRC Press, Boca Raton, FL, pp. 1279-1298. ISBN 978-1439829431

[5] Ericsson, A. and Simon, H. A. (1993) Protocol Analysis: Verbal reports as data. MIT Press, London, UK.

[6] Fernandez, A., Anrahao, S and Insfran, E (2013) Empirical validation of a usability inspection method for model-driven Web development Journal of Systems and Software, 68 (1) p161-186.

[7] Fox, M.C., Ericsson, K.A. & Best, R., 2011. Do procedures for verbal reporting of thinking have to be reactive? A meta-analysis and recommendations for best reporting methods. Psychological Bulletin, 137(2), pp.316-344

[8] Hertzum, M., Hansen, K.D. and Andersen, H.H.K. (2009). Scrutinising usability evaluation: does thinking aloud affect behaviour and mental workload? *Behaviour & Information Technology*. 28 (2). pp. 165-181

[9] Hertzum, M & Jacobsen, NE (2003) The Evaluator Effect: A Chilling Fact About Usability Evaluation Methods, International Journal of Human–Computer Interaction, 15:1, 183-204, DOI: 10.1207/S15327590IJHC1501\_14

[10] Holtzblatt, K., Wendell, J and Wood, S., (2004) Rapid Contextual Design. Morgan Kaufmann

[11] Hornbaek, K. (2010). Dogmas in the assessment of usability evaluation methods. Behaviour and Information Technology, 29(1), 97-111.

[12] Hornbaek, K and Frokjaer, E (2004) Two psychology-based usability inspection techniques studied in a diary experiment. NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction October 2004 Pages 3–12

[13] Hornbæk, K. and Frøkjær, E. (2008), “Comparison of Techniques for Matching of Usability-Problem Descriptions”, *Interacting with Computers*, 20(6), December 2008, Pages 505-514.ISO 2019

[14] Hornbæk, K and Frøkjær, E (2008) Making use of business goals in usability evaluation: an experiment with novice evaluators. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08). ACM, New York, NY, USA, 903-912.

[15]Hvannberg, Ebba Thora, Law, Effie Lai-Chong and Larusdottir, Marta Kristin (2007) Heuristic evaluation: Comparing ways of finding and reporting usability problems. Interacting with Computers Vol 19, no 2, p 225-240

[16] International Standards Organisation. (2019) ISO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems

[17] Jacobsen, N.E., Hertzum, M. and John, B.E. (1998) The Evaluator Effect in Usability Studies: Problem Detection and Severity Judgments. Proc. HFES 42ndAnnual Meeting. Human Factors and Ergonomics Society, SantaMonica, CA, pp. 1336–1340.

[18] Keinonen, T. (2009). Design method: Instrument, competence or agenda. Multiple Ways to Design Research: Research Cases Reshape the Design Discipline. M. Botta (ed). Swiss Design Network, 280-293

[19] Lavery, D Cockton, G, Atklnson, M.P. (1997): Comparison of Evaluation Methods Using Structured Usability Problem Reports. Behaviour and Information Technology, 16 (4) pp. 246-266

[20] McDonald, S., and Petrie, H. (2013) The effect of global instructions on think-aloud testing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Paris, France, April 27 - May 2. ACM New York, NY, USA. pp. 2941-2944

[21] McDonald, S., Zhao, T and Edwards, H.M. (2015) Look Who’s Talking: Evaluating the Utility of Interventions During an Interactive Think-Aloud. Interacting with Computers, doi:10.1093/iwc/iwv014

[22] Molich, R 2018. Are usability evaluations reproducible? Interactions 25(6), 82-85. DOI: https://doi.org/10.1145/3278154

[23] Monahan, K, McDonald, S and Cockton, G (2006) [*Modified Contextual Design as a Field Evaluation Method.*](http://sure.sunderland.ac.uk/id/eprint/36/) In: Fourth Nordichi conference on human computer interaction.

[24] Nielsen, J. 1992. Finding usability problems through heuristic evaluation. Proceedings ACM CHI'92 Conference (Monterey, CA, May 3-7), 373-380

[25] Sears, A and Hess, D. J. 1998. The effect of task description detail on evaluator performance with cognitive walkthroughs. In CHI 98 Conference Summary on Human Factors in Computing Systems (CHI '98). ACM, New York, NY, USA, 259-260.

[26] Schön, D.A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books

[27] Van Den Haak, M.J., De Jong, M.D.T., and Schellens, P.J. (2006) Constructive Interaction: An Analysis of Verbal Interaction in a Usability Setting, IEEE Transactions on Professional Communication, 49(4), 311-324.

[28] VanLehn, K and Jones, R.M. and Chi, M.T.H (1992) A model of the Self-explanation. Journal of the Learning Sciences. (1), 1-60

[29] Woolrych, A., Cockton, G. and Hindmarch, M., 2004 Falsification Testing for Usability Inspection Method Assessment. Proceedings of HCI 2004, Volume 2, eds. A. Dearden and Leon Watts, Research Press International, 137-140, 1-897851-13-8

[30] Woolrych, A, Hornbaek, K, Frokjaer, E and Cockton, G (2011) [*Ingredients and meals rather than recipes: a proposal for research that does not treat usability evaluation methods as indivisible wholes.*](http://sure.sunderland.ac.uk/id/eprint/1772/) International Journal of Human-Computer Interaction, 27 (10). pp. 940-970. ISSN 1044-7318

[31] Woolrych, A. (2011) *Understanding and Improving Analyst Decision Making in Usability Inspection* Unpublished PhD thesis, University of Sunderland, UK

[32] Zhao, T. McDonald, S., and Edwards, H.M. (2014) The impact of two different think-aloud instructions in a usability test: a case of just following orders? Behaviour and Information Technology, 33, 163–183