Evidence of Bias in the Production of User Test Lists by Software Analysts, and Proposed Mitigation Strategy

Leonel Morales[1], y Arturo Rivera[2]

[1]Universidad Rafael Landívar, Campus Central, Guatemala. [2]Escuela de Ingeniería, Universidad del Istmo, Guatemala.
[1]lmoralesd@url.edu.gt. [2]jarivera@unis.edu.gt.

Abstract. In the absence of sufficient HCI professionals, user test lists are generally developed by people with intimate knowledge of the software product in case. It has been seen that this may lead to bias when the language employed in the lists hints the user on how to perform the tests within the context of the application. This document presents empirical evidence of such bias and proposes a strategy to minimize it.

Keywords. Usability, User Tests

1 Introduction

There are abundant references to user tests as one of the most effective methods of identifying usability issues in software applications [9], [11].

The general practice consists of choosing a representative sample of the users to run the tests, and hand them a list of tasks developed by an experienced professional.

In our context, however, there is a marked shortage of HCI professionals, so the production of these lists is generally carried out by people who have in-depth knowledge of the software application to be tested, with instructions to include tasks that are representative of what users would normally do with the application. Such a list, however, will generally include, as we point out in this study, a series of clues and messages that the person who writes the list is sending surreptitiously to the participating users, thereby producing results which are biased or incomplete, and may even hide important issues from the evaluator

Such hints include application domain-specific terminology, distinctions or artificial classifications that may not be present in the mind of the ordinary user, or disguised indications that lead the user towards a specific menu option or link. These would provide information that would not be otherwise available and whose absence would make system use more difficult.

At first glance this may seem as an inherent limitation of user tests as a usability assurance technique, particularly because the specialist conducting the tests may not have enough knowledge about the application or application domain, which is

desirable in order to ensure neutrality, but at the same time may render him or her incapable of suggesting improvements to such lists.

As explained below, however, it is possible to take preventive and corrective measures to allow the effort and time invested in user tests yield the expected results.

This study seeks to provide empirical evidence of the issue, hints to identify it, and advice on how to handle it.

2 Importance of User Test Lists

There is abundant literature on the suitability and good results of user tests, [2], [3], [8], [9], [10], [11], there are methodological guides for their usage,[4], [6], metrics to use on tasks, etc., but we found few references regarding the aforementioned problem. Most studies deal mostly with issues like the number of users required to identify a given percentage of the issues, the probability that an issue will be identified, the specific moment within the software life cycle to perform the tests, their contribution to user-centered design processes and how they compare to other usability assurance techniques.

Several studies document usability evaluations using user tests for specific applications, [1], [5], [7], [13], [14], in different domains, some of them web-based, intranet-oriented, or more traditional desktop applications, usually without emphasis on how the task lists were developed.

Part of this is due to the diversity of functions and interactions that each application allows, their orientation, context, target users, etc., which makes the number of possible approaches to task list development potentially large and the construction process hardly amenable to generalization.

3 First Indications of Bias in the Development of User Test Lists from an Academic Setting

During the Software Engineering II course, taught by one of the authors during the second semester of 2006 at Universidad Rafael Landívar, the main topic was usability engineering of software applications.

The course presented several usability assurance techniques, and one of the assignments called for the students to select an application they had worked on, or were working in, in order to conduct user tests, and filming the test themselves. The task list that the participants where asked to carry out should be designed by themselves, and students were only required to present the taped tests to the rest of the class.

As was to be expected, the results revealed important usability problems, even in areas where the developers had not even suspected.

The results were also revealing to the instructor, because they showed that, in general, students had developed lists that made direct referentes to terminology, functions, buttons, menu items, etc. within the specific application, which evidently

provided artificial aid to users performing the tests, thereby invalidating the test or making it inefficient.

This behavior led us to wonder if the same problem would arise in the software industry if professional developers or integrators were writing the task lists for user tests.

4 Experiment on the Production of User Test Lists in Industry

Several software development companies from the Guatemala Export Software Commission (Sofex) were invited to participate in an exploratory study on this issue.

In the three companies that volunteered for the study, the authors held meetings with two-person teams, comprised of a developer and an integrator or analyst. The authors gave a brief introductory talk about usability, following the "5-e" approach (effective, efficient, eay to learn, error tolerant, engaging), as proponed by Quesenbery [12], and told that user tests on a specific application developed by the company would be performed, giving details about the technique, as well as other methods such as heuristic evaluation, inspection, direct observation, etc., and specifying the goals of the test. This was done in order to ensure that the lists would be written with the same frame of reference from one team to the other, and with sufficient context to understand the stated purpose.

At no point was it hinted that this was merely an exploratory or academic study, since this would have influenced the results. Instead, it was agreed with the directors of the participating companies that, at their option, the process could be continued with the selection of representative users and the actual performance of the tests.

5 Review of the Produced Lists

Participants were asked to submit their task lists by electronic mail in order to allow the authors to review them in digital format. Our first finding was that, just as was the case with students, there were obvious allusions, distinctions and other messages in the lists.

As an example, one of the lists, related to a CRM application, distinguishes between "personal" and "company" incidents. This separation may be considered artificial, as it is introduced by the application, possibly as a result of the classes in the object model, or the tables in the relational database model, so it may not be justified to assume that the user knows and identifies them clearly. This would have introduced an additional degree of complexity if the user attempts to record a "generic" incident, because he or she would be forced to determine, from the user interface, that the system makes a distinction between the two types of incidents.

Since this study was merely exploratory, it is not possible to generalize the findings, or attempt to identify all the possible issues. For illustrative purposes, the following table shows actual excerpts from the task lists, indicating some of the deficiencies in them.

Statement in Task List	Comments
Creation of personal incidents. Creation of company incidents.	This suggests that there is a distinction in the system between two types of incidents, which may not be natural to the user. It also does not provide concrete data for performing the test. The terminology may be specific to the application.
Enter a product family. Enter a product sub-family. Enter a product sub-sub-family.	Analogous to the previous case, suggests that there is a finite, three-tier hierarchy, which always require all three levels. This may or may not coincide with the users' expectations or mental model.
Mark the checklist. Enter the system / Exit the system. Save changes and close.	These are typically meaningless tasks for the user, that is, it is not something that the user seeks to do as part of his or her job, but rather a necessity imposed by the use of the system.
Check the time measurement report. Query income and cost.	The user's objective will hardly be to perform a query or report. It is more likely to be obtaining a specific set of data, the report or query being the means (potentially not the only one) to achieve it. The task statement suggest the path preferred by the designer.
1. Insert text "xxxx". 2. Insert text "yyyy". 3. Add the background image in folder C:\Carnet.	Besides the fact that the user's objective is not inserting the elements, the suggested order is artificial.
Read messages. Send messages.	These tasks, besides being a means to an end, have been expressed in a highly generic fashion, being meaningless to the user by themselves, and not providing the information that may be required to execute them.

6 Proposed Mitigation Strategy and Results

Considering the aforementioned issues, the authors reflected about a strategy that might be useful in order to mitigate or even avoid them.

Naturally, an easy approach would be to simply warn the people in charge of developing the list about the potential consequences of these mistakes, explaining the need to avoid them and even suggesting better ways to state the tasks. It might be

worthwhile to include some of this in the introductory material for those writing the test lists. However, there is a potentially large number of defects to avoid, as suggested by the fact that this scaled-down study yielded so many different types of issues. This would make the induction process unbearably complex, and generate confusion by the number of cases to consider. A more general strategy is desirable. Maybe some of these issues can only be fully solved by the involvement of many more HCI professionals, who would naturally avoid the aforementioned mistakes. This, however, is not a viable short term solution, so it is important to seek a more general approach that minimizes the negative effects of bias.

As a result of the joint reflection of the authors, and borrowing from ideas in one of the suggested task lists, it was considered that a promising approach would be to ask the writers to present the tasks as a series of stories or cases, presenting the users with situations that they are likely to find in real life, emphasizing the context and objective of task rather than the task itself. The more complete and realistic the scenario, the greater the chance that the tests would give valid and useful results. For example, for an ID-issuing application, the task could be specified giving a sample of the desired design and asking the user to reproduce it within the application.

7 Conclusions and Next Steps

As indicated, the present study has significant limitations regarding the size of the sample. A wider study is required in order to generalize the results. Additionally, it would be very important to validate the proposed strategy in order to objectively assess its virtues and limitations. However, the results so far are promising and justify the continuation of this work.

Some preliminary conclusions are suggested by this study. First, it was noted that analysis exhibit a tendency to think in terms of abstract, general cases, while tests require concreteness and specificity. This could be aggravated by the fact that the systems for which the task lists were developed fall into the COTS[1] category, since they are intended to appeal to a wider market. This, in time, points to intrinsic obstacles for this kind of tests to be developed independently (something that would have been desirable, and is strongly recommended byt some authors, particularly [10]), which would requiere an HCI specialist for them, something that is not feasible in the current context. Also, it became evident that the writers possess a conceptual model [11] of the application that reflects the way in which it was designed, and so the task lists attempt to impose this model to the user, regardless of the way he or she perceives daily tasks, thereby negating the purpose of the test.

However, the produced lists do provide evidence of tasks, such as exiting the application, printing a report or changing the password which, even though artificial, are widely accepted as a necessity of the application or its environment, and as such need to be tested.

[1] Commercial Off-The Shelf

References

1. Arbildi, I.: Caso de Estudio: Técnicas de Arquitectura de Información Aplicadas al Desarrollo del Sitio Web de Ibai Intranets. In: El Profesional de la Información. Vol. 13. No. 3. (2004)
2. Baeza-Yates, R., Rivera, C., Velasco, J.: Arquitectura de la Información y Usabilidad en la Web. In: El Profesional de la Información, Vol. 13, No. 3. (2004)
3. Cover, D.: Usage and Usability Assesment: Library Practices and Concerns. Digital Library Federation, Council on Library and Information Resources. (2002)
4. Dix, A., Finlay, J., Abowd, G., Beale, R.: Human-Computer Interaction. 2nd edn. Prentice Hall. (1998)
5. James, R., McDonald, A., McGuire, R.: A Usability Evaluation of a Home Monitoring System. Symposium on Usable Privacy and Security (SOUPS). (2007)
6. Laurel, B. (ed) : The Art of Human Computer Interface Design. Adison-Wesley. (1990)
7. Marcos, M., Rovira, C.: Evaluación de la Usabilidad en Sistemas de Información Web Municipales: Metodología de Análisis y Desarrollo. 7mo Congreso ISKO-España. (2005) 415-432
8. Montes de Oca, A.: Arquitectura de Información y Usabilidad: Nociones Básicas para los Profesionales de la Información. Acimed Vol. 12, No. 4. (2004)
9. Nielsen, J.: Usability 101: Introduction to Usability. In: Jakob Nielsen's Alertbox, August 25. (2003)
10. Nielsen, J.: Misconceptions About Usability. In: Jakob Nielsen's Alertbox, September 8. (2003)
11. Norman,D.: The Design of Everyday Things. Basic Books. (1988)
12. Quesenbery, W.: What Does Usability Mean: Looking Beyond 'Ease of Use'. Proceedings of the 48th Annual Conference, Society for Technical Communication. (2001)
13. Whitten, A., Tygar, J.: Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. Proceedings of the 8th USENIX Security Symposium. (1999)
14. Withrow, J., Brinck, T., Speredelozzi, A.: Comparative Usability Evaluation for an e-Government Portal. Diamond Bullet Design Report #U1-00-2. (2000)