Evidencia de Sesgo en la Elaboración de Listas de Pruebas de Usuario por Parte de Analistas y Estrategia de Mitigación

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. En ausencia de suficientes profesionales de la usabilidad, las listas para pruebas de usuario son generalmente desarrolladas por personas con íntimo conocimiento del software en cuestión. Se ha observado que esto puede generar sesgo cuando el lenguaje de las listas sugiere cómo desarrollar las pruebas dentro del contexto de la aplicación. En este documento se presenta evidencia empírica de este sesgo, y se propone una estrategia para reducir el mismo.

Keywords. Usabilidad, Pruebas de Usuario

1 Introducción

Las referencias a las pruebas de usuario como uno de los métodos más efectivos para identificar problemas de usabilidad en aplicaciones de software son abundantes [9], [11].

La práctica general consiste en elegir una muestra de usuarios representativos para las pruebas y luego entregarles una lista de tareas elaborada para el efecto por un profesional con amplia experiencia en el tema.

En nuestro medio existe una marcada escasez de profesionales calificados en temas de Interacción Humano-Computador, por lo que la elaboración de la lista de tareas usualmente se encarga a una persona o equipo de personas que conocen la aplicación o software que será probado, encomendándoles que traten de incluir tareas representativas de las que los usuarios a menudo deben realizar. Una lista producida de esta forma muy probablemente incluirá, tal y como lo hacemos ver en este estudio, una serie de mensajes y pistas que quien elabora la lista envía subrepticiamente al usuario que participa en las pruebas, produciendo resultados sesgados, incompletos o incluso ocultando problemas importantes al evaluador.

Términos específicos del dominio de la aplicación, distinciones o clasificaciones artificiales que en la mente del usuario no necesariamente están presentes, indicaciones veladas que guían hacía una determinada opción de menú o enlace, etc.,

pueden incluirse en la lista de tareas y proveer de información que de otra forma no estaría disponible y cuya carencia haría más difícil el uso del sistema.

A primera vista puede parecer que este problema es una limitación inherente de las pruebas de usuario como técnica de aseguramiento de la usabilidad, especialmente porque el especialista que conduce las pruebas puede no tener suficiente conocimiento del software o siquiera del dominio de la aplicación, lo cual es aconsejable para garantizar que el usuario se mantenga ajeno a cualquier información adicional que le ayude a realizar las tareas incluidas en la lista, pero por esto mismo sentirse incompetente para hacer modificaciones o correcciones en la redacción presentada.

Sin embargo, como se indica más adelante, es posible tomar medidas de prevención e incluso medidas correctivas, para evitar que el esfuerzo y tiempo invertidos en las pruebas de usuario se pierdan o no rindan todos los beneficios que se esperaban.

En este estudio buscamos aportar evidencia empírica del problema, indicaciones para identificarlo y consejos para enfrentarlo.

2 Importancia de las Listas de Pruebas de Usuario

La literatura describe abundantemente la conveniencia y buenos resultados de las pruebas de usuario, [2], [3], [8], [9], [10], [11], se dispone de guías metodológicas para su realización,[4], [6], métricas a emplear sobre las tareas, etc., pero encontramos pocas referencias cercanas al problema señalado. Los estudios se centran sobre todo en aspectos como el número de usuarios necesario para identificar un porcentaje establecido de problemas, la probabilidad de que un problema sea identificado, el momento en el ciclo de vida del software en que es adecuado realizarlas, su peso y aporte en procesos de diseño centrado en el usuario, y cómo se compara con otras técnicas de aseguramiento de la usabilidad.

Múltiples estudios documentan evaluaciones de usabilidad con pruebas de usuario en aplicaciones concretas, [1], [5], [7], [13], [14], de diferentes dominios, algunas basadas en la web, una intranet o bien ejecutables más tradicionales, a menudo sin hacer ningún énfasis especial en cómo se generó la lista de tareas.

En parte esto se debe a la diversidad de funciones e interacciones que cada aplicación permite realizar, su orientación, contexto, usuarios objetivo, etc., lo que hace que la cantidad de formas en que una lista de tareas puede elaborarse sea extensa y el proceso de su construcción difícilmente generalizable.

3 Indicios de Sesgo en la Elaboración de Listas de Pruebas de Usuario desde el Ámbito Académico

El curso de Ingeniería de Software II, impartido por uno de los autores durante el segundo semestre del año 2006 en la universidad Rafael Landívar, se concentró en el estudiar la ingeniería de la usabilidad de aplicaciones de software.

En el contenido se presentaron las diferentes técnicas de aseguramiento de la usabilidad y se pidió a los estudiantes que eligieran una aplicación en la que

estuvieran trabajando o hubieran trabajado, para conducir pruebas de usuario, filmando la prueba en sí. La lista de tareas que los participantes debían completar con el software debía ser diseñada por ellos mismos y sólo se les requirió presentar el documento en video al resto de la clase.

Como era de esperarse, los resultados revelaron problemas importantes de usabilidad en puntos que los desarrolladores no habían siquiera sospechado que podían existir.

Adicionalmente también fueron reveladores para el profesor por mostrar que, en general, los alumnos habían elaborado listas que hacían referencias directas a terminología, funciones, botones, opciones de menú, etiquetas de enlace, etc., de las aplicaciones concretas, que evidentemente prestaban una ayuda artificial al usuario que realizaba las tareas, invalidando o desvirtuando la prueba.

Este comportamiento nos llevó a preguntarnos si en la industria podía presentarse el mismo problema si se pedía a los equipos de desarrolladores o integradores profesionales que elaboraran listas de tareas para ser utilizadas en pruebas de usuario.

4 Experimento en Elaboración de Listas de Pruebas de Usuario en la Industria

Se invitó a diversas empresas de desarrollo de software y sistemas, afiliadas a la Comisión de Software de Guatemala a participar en un estudio exploratorio para el efecto.

En las tres empresas voluntarias, se realizaron reuniones con equipos de dos personas, un desarrollador y un integrador de sistemas, a quienes se les dio una charla introductoria sobre usabilidad usando el enfoque de las cinco “e” de usabilidad (effective, efficient, easy to learn, error tolerant, engaging) según lo propone Quesenbery [12] y luego se les explicó que se harían pruebas de usuario de una aplicación específica desarrollada por ellos, dándoles los detalles de la técnica, lo mismo que las de otras como evaluación heurística, por inspección, observación directa, etc., y especificando los objetivos de las pruebas. Esto para garantizar que las listas eran elaboradas con el mismo marco de referencia de un equipo a otro y con el contexto suficiente para entender lo que se proponía.

En ningún momento se les indicó que se tratara sólo de un estudio exploratorio o de una investigación académica, pues esto podía influenciar el resultado, y se acordó con los directivos de las empresas participantes que, si ellos lo decidían, se podía continuar con la selección de los usuarios representativos y llevar a término las pruebas.

5 Revisión de las Listas

Se indicó a los participantes que enviaran sus listas de tareas por correo electrónico, por lo que dispusimos de versiones digitales desde el principio. Nuestro primer hallazgo fue encontrar lo evidente que las alusiones, distinciones y otros mensajes, resultaban dentro de las listas.

Por ejemplo, en una de las listas, referida a una aplicación CRM, se hace la distinción entre "incidentes de personas" e "incidentes de empresas". Esta distinción puede considerarse artificial e introducida por la aplicación (probablemente derivada de las clases implementadas en el modelo de objetos o bien de la organización de tablas en el modelo relacional de la base de datos) por lo que no se puede asumir que el usuario la conoce e identifica claramente, lo que le habría puesto una barrera extra al momento de querer ingresar un incidente "genérico", pues primero habría tenido que establecer, usando sólo el entorno de la interfaz de usuario, que el sistema hace la distinción entre los dos tipos de incidentes.

Debido a que nuestro estudio fue de carácter exploratorio únicamente, no nos es posible generalizar los resultados ni pretender que se identifican todos los problemas posibles. Para fines ilustrativos se presenta la siguiente tabla, con ejemplos de algunos enunciados de las listas de pruebas, señalando las deficiencias de las mismas.

Enunciado en la Lista de Tareas	**Comentario**
Creación de incidentes personas Creación de incidentes empresa	Sugiere que hay una distinción en el sistema entre ambos tipos de incidentes que puede no ser natural para el usuario. Además no provee datos concretos para la prueba. La terminología empleada puede ser propia del sistema.
Ingresar una familia de productos Ingresar una sub-familia de productos Ingresar una sub-sub-familia de productos	Análogo al caso anterior, sugiere que ésta es una jerarquía finita, y que requiere siempre de los tres niveles, lo cual puede o no coincidir con la expectativa o el modelo mental del usuario.
Marcar checklist Ingresar al sistema / Salir del sistema Guardar los cambios y cerrar	Se trata de tareas que carecen de propósito para el usuario, es decir, no son algo que un usuario desee hacer como parte de su trabajo, sino una necesidad impuesta por el uso del sistema.
Consultar reporte de medición de tiempos Realizar una consulta de los ingresos con costo	El objetivo del usuario difícilmente será realizar una consulta. En todo caso perseguirá obtener un dato específico, siendo el reporte un medio (probablemente no el único) para lograrlo. El enunciado de la tarea sugiere el mecanismo preferido por el diseñador.
1. Insertar el texto "xxxx" 2. Insertar el texto "yyyy" 3. Agregar una imagen de fondo que se encuentra en el directorio C:\Carnet	Además de que el objetivo del usuario no sería insertar estos elementos, el orden sugerido es artificial.
Leer mensajes Enviar mensajes	Estas tareas, además de que son un medio para un fin, están expresadas en forma genérica y no dicen nada al usuario en sí mismas, ni acerca de la información que

	podría requerirse para realizarlas.

6 Propuesta de Estrategia de Mitigación y Resultados

Considerando los inconvenientes anteriormente anotados, se reflexionó acerca de una estrategia que pudiera resultar útil para mitigarlos, o incluso evitarlos.

Naturalmente, una forma fácil de atacar el problema sería simplemente advertir a los encargados de elaborar las listas de los tipos de inconvenientes que pueden generar, explicando por qué es necesario evitarlos, pudiéndose dar incluso algunos ejemplos de redacciones más apropiadas. Probablemente sería conveniente incluir algo de este material en el proceso de inducción para quienes redactan las pruebas. Sin embargo, existe potencialmente una gran variedad de defectos que deba evitarse, como sugiere el hecho de que un estudio tan limitado como el presente haya descubierto problemas de muchos tipos. Esto podría hacer inmanejablemente compleja la inducción, y generar confusión ante el número de casos a considerar, haciendo deseable el aplicar una estrategia más general. Quizás estos problemas solamente puedan ser resueltos en su totalidad a través de la introducción de más profesionales especializados en usabilidad, que por tanto evitarían los defectos citados. Sin embargo, mientras que esto no sea factible en el corto plazo, resulta conveniente buscar una estrategia alternativa que minimice los efectos negativos del sesgo.

Como resultado de la deliberación conjunta de los autores, y tomando ideas de una de las pruebas sugeridas en las listas, se consideró que una propuesta prometedora sería el pedir a los redactores que presenten las tareas en forma de historias o casos que planteen a los usuarios situaciones que puedan encontrar en la vida real, fijándose más bien en el contexto y el objetivo de la tarea antes que en la descripción de la tarea misma. Mientras más completo y realista sea el escenario, mayor probabilidad habría de que las pruebas arrojen resultados válidos y útiles. Por ejemplo, para una aplicación de generación de identificaciones, la tarea podría enunciarse dando una muestra del diseño buscado y pidiendo que se reproduzca en el programa.

7 Conclusiones y Siguientes Pasos

Como se ha indicado, el presente estudio tiene limitaciones significativas por el tamaño de la muestra. Es necesario realizar un estudio más amplio para poder generalizar los resultados. Adicionalmente, sería importante validar la estrategia de corrección propuesta, para poder identificar objetivamente sus virtudes y limitaciones. Sin embargo, lo obtenido hasta el momento resulta prometedor y justifica el continuar este trabajo.

Como producto del estudio se sugieren algunas posibles conclusiones preliminares. En primer lugar, se notó que los analistas e implementadotes tienen una tendencia a pensar en casos abstractos, generales, mientras que las pruebas requieren concreción y especificidad. Esto pudiera agravarse por el hecho de que los sistemas para los cuales

se desarrollaron las listas son de tipo COTS[1], por apelar a un mercado más amplio. Esto a su vez apunta a que existen obstáculos intrínsecos para que este tipo de pruebas pueda ser desarrollado en forma independiente por los mismos equipos que desarrollan el software, sin contratar especialistas en usabilidad (algo que habría sido deseable e incluso se recomienda por algunos autores, notablemente [10]), contratación que de todas formas, tal y como se explicó antes, en el contexto actual no resulta viable. Es evidente que los redactores cuentan con un modelo conceptual [11] de la aplicación correspondiente a la forma como ésta fue diseñada, y en la lista de tareas imponen el mismo al usuario, dejando de lado el que éste pueda tener respecto a sus tareas cotidianas, lo que en esencia desvirtúa el propósito de la prueba.

Sin embargo, las listas elaboradas sí aportan evidencia de tareas, como salir del sistema, imprimir reporte o cambiar contraseña que, aunque sean artificiales, son aceptadas como una necesidad del programa o su entorno, por lo que deben ser probadas.

Referencias

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)

[1] Commercial Off-The Shelf