

PROPUESTA PARA INCORPORAR EVALUACIÓN Y PRUEBAS DE USABILIDAD DENTRO DE UN PROCESO DE DESARROLLO DE SOFTWARE

BEATRIZ E. FLORIÁN*
OSWALDO SOLARTE**
JAVIER M. REYES***

RESUMEN

La usabilidad es crítica para el éxito de los sistemas de software interactivos. Las pruebas y evaluaciones de usabilidad durante el desarrollo del producto han ganado amplia aceptación como estrategia para mejorar la calidad del producto. La introducción temprana de las perspectivas de usabilidad en un producto es muy importante para brindar una clara visibilidad de aspectos de calidad, tanto para los desarrolladores como los usuarios de pruebas. Sin embargo, la evaluación y pruebas de usabilidad no es común que se tomen en cuenta como elementos indispensables del proceso de desarrollo de software. Este artículo expone una propuesta para introducir la evaluación y pruebas de usabilidad dentro de un desarrollo de software, basándose en la reutilización de artefactos de software. Adicionalmente, propone la introducción de un auditor dentro de la clasificación de actores para las pruebas de usabilidad y una mejora de las listas de chequeo utilizadas para evaluación heurística, agregándoles aspectos cuantitativos y cualitativos.

PALABRAS CLAVE: desarrollo de software; interacción humano-computador; pruebas de usabilidad; evaluación de usabilidad; evaluación heurística; listas de chequeo heurísticas.

* Ingeniera de Sistemas, Universidad del Valle. Magíster en Ingeniería de Sistemas y Computación, Universidad de los Andes. Estudiante de Doctorado, Universitat de Girona, España. Docente, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle. Cali, Colombia. beatriz.florian@correounivalle.edu.co; bflorian@eia.udg.edu

** Ingeniero de Sistemas y Magíster (c) en Ingeniería de Sistemas, Universidad del Valle. Docente, Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle. Cali, Colombia. osolarte@univalle.edu.co

*** Diseñador Industrial, Universidad Industrial de Santander. Especialista en Diseño de Ambientes de Aprendizaje. Magíster (c) en Ingeniería de Sistemas, Universidad del Valle. Docente, Departamento de Diseño, Facultad de Artes Integradas, Universidad del Valle. Cali, Colombia. javier.reyes@correounivalle.edu.co

PROPOSAL FOR INTRODUCING USABILITY EVALUATION AND TESTING WITHIN A SOFTWARE DEVELOPMENT PROCESS

ABSTRACT

Usability is critical to consider an interactive software system successful. Usability testing and evaluation during product development have gained wide acceptance as a strategy to improve product quality. Early introduction of usability perspectives in a product is very important in order to provide a clear visibility of the quality aspects not only for the developers, but also for the testing users as well. However, usability evaluation and testing are not commonly taken into consideration as an essential element of the software development process. Then, this paper exposes a proposal to introduce usability evaluation and testing within a software development through reuse of software artifacts. Additionally, it suggests the introduction of an auditor within the classification of actors for usability tests. It also proposes an improvement of checklists used for heuristics evaluation, adding quantitative and qualitative aspects to them.

KEY WORDS: software development; human-computer interaction; usability testing; usability evaluation; heuristic evaluation; heuristics checklists.

PROPOSTA PARA INCORPORAR A AVALIAÇÃO E PROVAS DE USABILIDADE DENTRO DE UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

RESUMO

A usabilidade é crítica para o sucesso dos sistemas de software interativos. As provas e avaliações de usabilidade durante o desenvolvimento do produto têm ganhado ampla aceitação como estratégia para melhorar a qualidade do produto. A introdução adiantada das perspectivas de usabilidade em um produto é muito importante para brindar uma clara visibilidade de aspectos de qualidade tanto para os desenvolvedores como os usuários de provas. No entanto, a avaliação e provas de usabilidade não é comum que se tomem em conta como elementos indispensáveis do processo de desenvolvimento de software. Este artigo expõe uma proposta para introduzir avaliação e provas de usabilidade dentro de um desenvolvimento de software, baseando-se na reutilização de artefatos de software. Adicionalmente, propõe a introdução de um auditor dentro da classificação de atores para as provas de usabilidade e uma melhoria das listas de revisão utilizadas para avaliação heurística, acrescentando-lhes aspectos quantitativos e qualitativos.

PALAVRAS-CÓDIGO: desenvolvimento de software; interação humano-computador; provas de usabilidade; avaliação de usabilidade; avaliação heurística; listas de revisão heurísticas.

1. INTRODUCCIÓN

La usabilidad ha sido considerada un atributo de calidad del software determinante para el éxito de un proyecto, generándole un interés creciente en el mundo del desarrollo de software como factor de calidad (Ferré, 2003; Cheikhi, Abran y Suryan, 2006). Las valoraciones de usabilidad en software se han realizado desde 1971. Para el año 2003 el *National*

Institute of Standards and Technology (NIST) identificaba más de 30 técnicas para conducir valoraciones de usabilidad (Ramli y Jaafar, 2008). Actualmente, la usabilidad es considerada como un atributo de la calidad del uso del software (ISO/IEC, 2009). Esta nueva definición permite hacer medidas más precisas sobre la usabilidad de un producto de software (Bevan, 2009).



Los métodos de valoración de usabilidad pueden dividirse en tres grupos: métodos de análisis, métodos de inspección y métodos de indagación. En los primeros los usuarios representativos trabajan en tareas típicas utilizando el sistema o prototipo, los evaluadores se concentran en observar como la interfaz posibilita en los usuarios realizar las tareas. En los segundos, se concentran en evaluar la interfaz por parte de especialistas en usabilidad o desarrolladores de software. En los terceros, los evaluadores se concentran en obtener información respecto a los gustos, disgustos, necesidades y comprensión del sistema de parte de los usuarios.

Desde el punto de vista de los participantes, se podrían resumir estos tres métodos en dos grandes grupos: *métodos no empíricos* y *métodos empíricos*. Aquéllos implican la participación de expertos especialistas en usabilidad, y éstos se conciben como instrumentos de análisis que requieren la participación de usuarios. En esta investigación se ha adoptado esta segunda clasificación, ya que abarca la anterior de manera más coherente para los intereses que se abordaron en las pruebas.

En la valoración de usabilidad de software, tanto las evaluaciones heurísticas (González, Lorés y Pascual, 2001; Nielsen, 1993) como las pruebas de usuario (Nielsen, 1993) generan un puente entre las ideas que tienen los desarrolladores sobre la interfaz y las ideas de los usuarios. La evaluación heurística es un método no empírico (evaluación de expertos) (Rubin y Chisnell, 2008), mientras que las pruebas de usabilidad se basan en métodos empíricos (pruebas con usuarios). El objetivo de ambas es realizar tareas que arrojen realimentación a los desarrolladores para depurar eficazmente la interfaz de usuario (Rosenbaum, 1989). Una de las grandes ventajas de la evaluación heurística es que no requiere una larga planificación y que puede usarse desde las etapas iniciales del proceso de desarrollo del sistema con los mismos desarrolladores de la aplicación (González, Lorés y Pascual, 2001).

En particular, los acercamientos iniciales de Nielsen (1999) y de Nielsen y Loranger (2006) al respecto de las evaluaciones heurísticas en la web arrojan una serie de principios heurísticos como requisitos mínimos para el diseño de interfaces web más usables y accesibles. Estos principios se toman como punto de partida para las evaluaciones desarrolladas en esta investigación, junto con los de Tidwell (2006) y LabIUtil (2003), González, Lorés y Pascual (2001) y Shneiderman y Plaisant (2006).

A pesar de las ventajas, la evaluación y pruebas de usabilidad, por lo general, no son tomadas en cuenta como elementos indispensables del proceso de desarrollo de software. Algunas propuestas en torno a este planteamiento se encuentran en Hakiel (1997), Cysneiros y Kushniruk (2003), Tao (2005), Singh (2008) y una paralela a nuestro trabajo en Aveledo y De la Rosa (2010). Hakiel (1997) habla de dos problemas relacionados con la usabilidad en el desarrollo de software; el primero es que los requisitos sólo tienen en cuenta la ingeniería del producto, y el segundo se refiere a que no se tienen en cuenta los factores humanos en el proceso de desarrollo; el autor plantea una serie de actividades orientadas a la usabilidad a través de las etapas del desarrollo de software; sin embargo, sólo hace referencia a las actividades asociadas en cada etapa, pero no propone artefactos o mecanismos concretos que ayuden a evaluar la usabilidad. Cysneiros y Kushniruk (2003) se enfocan sólo en solucionar las posibles interpretaciones de usabilidad por medio de la construcción de un catálogo de conceptos relacionados. Este catálogo se usa para construir los requisitos de usabilidad del proyecto. Tao (2005) propone un modelo basado en estados de máquina y heurísticas de usabilidad. Los estados de máquina permiten representar la interacción del usuario con el sistema, y las heurísticas de usabilidad se aplican para mejorar el diseño de las interfaces de usuario. Este modelo se enfoca en mejorar la formación en usabilidad para aplicarla al proceso de desarrollo de software. Singh (2008) extiende la metodología ágil Scrum incluyendo la usabilidad en el proceso; a

esta propuesta la llama U-SCRUM. El autor plantea la necesidad de tener dos personas encargadas del producto: el responsable de la funcionalidad y el responsable de la usabilidad. No obstante, se debe tener cuidado al aplicar esta metodología, ya que los dos responsables mencionados podrían entrar en desacuerdo, si no se trabaja el desarrollo del producto como un objetivo común. Los tipos de artefactos que propone Singh (2008) son: los roles de usuario, la visión de experiencia de usuario y el plan del producto; en esta propuesta no se menciona cómo se evaluarán las características de usabilidad ni un mecanismo para verificarlas.

Haciendo una mejora de los trabajos anteriores, el objetivo principal de este estudio es proponer actividades y artefactos que enriquezcan el proceso de desarrollo de software a partir de la introducción temprana de requisitos de usabilidad en un producto; las tareas la permiten y los artefactos entregables ayudan a verificar que esas tareas se cumplan continuamente en el proceso. En ese sentido este estudio continúa las propuestas de Granollers *et al.* (2005), de Juristo, Moreno y Sánchez-Segura (2007) y de Avello, De la Rosa y Moreno (2008). En la propuesta se relacionan evaluaciones heurísticas con pruebas de usuarios, conectando los artefactos propuestos, tanto de las evaluaciones como de las pruebas. La idea principal es combinar las plantillas de evaluación heurísticas realizadas por expertos, para que también sirvan de guía a los usuarios durante las pruebas de usabilidad. Con esta propuesta se brinda una clara visibilidad de los aspectos de usabilidad para los desarrolladores y para los usuarios de pruebas desde etapas tempranas del desarrollo. La propuesta está pensada para evitar impactar negativamente la duración y costos del proyecto.

Dentro de esta investigación también se sintetiza la clasificación de actores, propuesta por diferentes autores, para las pruebas de usabilidad y se propone la figura del *auditor* dentro de esta clasificación.

Florián *et al.* (2007) presentan actividades y artefactos que se usaron durante el desarrollo de la plataforma computacional PREDICA (Plataforma Experimental para Sistemas de Recomendación, Descubrimiento de Conocimiento, Interfaces Adaptativas y Consultas Avanzadas). El propósito del proyecto PREDICA fue desarrollar una plataforma experimental para facilitar la búsqueda de documentos en el área de la computación cuya interfaz se adapta a un modelo de usuario definido y que ofrece recomendaciones con base en un perfil de consulta. Las interfaces de PREDICA son evaluadas y probadas dentro de este estudio, en el cual las pruebas con usuarios descritas se complementan con otras basadas en la *técnica de análisis de tarea y actividad* (Kafure, 2000 y 2004; Medeiros, Kafure y Lula, 2000), documentadas en Kafure *et al.* (2007).

El artículo está distribuido de la siguiente manera. En la sección 2 se presentan la clasificación de actores para las pruebas y evaluaciones heurísticas de usabilidad y la propuesta del actor *auditor*. En la sección 3 se describen las actividades y artefactos propuestos. En la sección 4 se describe la aplicación del modelo propuesto al desarrollo de la biblioteca digital PREDICA. En la sección 5 se exponen los resultados de investigación. Finalmente, se presentan las conclusiones y referencias bibliográficas.

2. CLASIFICACIÓN DE ACTORES DE PRUEBAS

De acuerdo con los planteamientos de Mayhew (1999), Granollers, Lorés y Cañas (2005) y Alarcón *et al.* (2007), el usuario forma parte del proceso de desarrollo en diversas etapas. Uno de los conceptos fundamentales en este sentido es la iteración en el proceso de evaluación, donde se pueden realimentar los prototipos funcionales con base en criterios estructurados de usabilidad (Shneiderman y Plaisant, 2006).



De las estrategias de clasificación de usuarios planteadas por Mayhew (1999) y Granollers, Lorés y Cañas (2005) se utilizaron dos: la estrategia de conocimiento y experiencia, que se basa en la frecuencia de uso de las herramientas computacionales, y la estrategia de estructuración de las tareas de interacción, basada en los modelos mentales de los usuarios con respecto al uso de las tecnologías en su trabajo. La clasificación responde a la necesidad de categorizar a los usuarios dentro de unos niveles de desempeño, porque esto facilita los procesos de evaluación de las interfaces de usuario (Shneiderman y Plaisant, 2006).

Utilizando las estrategias anteriores se presentan cinco grupos de usuarios de prueba. Dos de los grupos involucran expertos en usabilidad, el *auditor* propuesto en este artículo y el *desarrollador* propuesto por Aveledo y Moreno (2008) y los otros tres corresponden a diferentes tipos de usuarios de la aplicación de software desarrollada propuestos por Shneiderman y Plaisant (2006). Los grupos de usuarios de prueba de esta clasificación podrían adecuarse a cualquier tipo de aplicación de software que se quiera adelantar. A continuación se presentan los cinco grupos de clasificación de usuarios.

Usuario novato o inexperto. Usuario que tiene poco conocimiento de las herramientas computacionales y cuya interacción con aplicaciones similares a la que se quiere construir no es frecuente. Dedicar entre un 0 % y 20 % de sus actividades a tareas similares a las que se realizarán con el producto de software por construir.

Usuario intermedio. Es aquel usuario que utiliza con frecuencia el computador. Dedicar entre un 20 % y 80 % de sus actividades a tareas similares a las que se ejecutarán con el producto de software por construir.

Usuario avanzado. Este usuario ocupa mucho tiempo interactuando con herramientas computacionales, que suele usarlas por razones de trabajo. El 80 % o más de sus actividades involucran tareas similares a las que se llevarán a cabo con el producto de software.

Desarrollador. Se propone al desarrollador como un integrante más del proceso de evaluaciones de usabilidad. Para evitar las observaciones técnicas que se alejan un poco de los criterios de interacción humano-computador (IHC) se han utilizado listas de chequeo que dan las pautas para que los desarrolladores tengan la posibilidad de observar más de cerca la interacción.

Auditor. Es quien realiza la verificación del sistema desde el punto de vista funcional, pero teniendo en cuenta criterios de usabilidad. Por tanto, es el experto en usabilidad planteado por González, Lorés y Pascual (2001), que además debe tener conocimientos en ingeniería de software y en las tecnologías utilizadas en el desarrollo del producto. El auditor no participa en las etapas de análisis e implementación del software. Provee una visión más holística del proceso de interacción, pues se encarga de evaluar, con base en unas heurísticas claras planteadas con anterioridad al proceso mismo de la evaluación. Esto permite que la evaluación se desarrolle de una manera menos subjetiva y evita la posibilidad de tener apreciaciones que estén descontextualizadas respecto a la interacción de los usuarios potenciales con la aplicación.

3. PROPUESTA DE TAREAS Y ARTEFACTOS DE USABILIDAD DENTRO DEL DESARROLLO DE SOFTWARE

Esta propuesta incluye tareas de pruebas de usabilidad desde etapas tempranas del desarrollo de software y se aleja de la visión de que sólo corresponden a las etapas finales o de transición. De esta manera busca cambiar la concepción de esperar a que el producto de software esté construido para indagar en él los aspectos de usabilidad y, en cambio, propone anticiparse, para tener en cuenta aspectos de usabilidad desde las etapas de levantamiento de requisitos y diseño del software para luego realizar las ejecuciones de pruebas y evaluaciones de usabilidad y finalizar con la verificación de la interfaz.

La propuesta está pensada para grupos de desarrollo medianos, donde los desarrolladores están familiarizados con la realización de tareas de pruebas durante la implementación. Se debe capacitar previamente a los desarrolladores en el área de usabilidad del tipo de productos que crean, con el fin de que puedan cumplir con el papel de expertos en usabilidad. Como los desarrolladores conocen los principios de usabilidad antes de la construcción de la interfaz, esto les permite tenerlos en cuenta en el proceso de implementación (Ferré, 2003).

Pensando en lograr la reutilización de artefactos, esta propuesta plantea basar la ejecución de pruebas y evaluaciones heurísticas de usabilidad principalmente en un tipo de artefacto: las listas de chequeo mejoradas. Las listas de chequeo básicas están creadas a partir de un conjunto de heurísticas (en el caso de este proyecto las denominaremos características) de usabilidad recopiladas de las propuestas de Nielsen (1993), Tidwell (2006) y LabiUtil (2003), González, Lorés y Pascual (2001) y Shneiderman y Plaisant (2006). Los expertos en usabilidad aportarán otras características a las listas de chequeo según la aplicación de software particular que se desarrollará. Estas características sirven como material de referencia para realizar una evaluación consistente y objetiva. Las mejoras que se apliquen a las listas en esta propuesta se explicarán al final de esta sección. La reutilización de la lista de chequeo es una herramienta importante a la hora de evaluar, pues esto ayudó a centralizar las observaciones en aspectos que tenían que ver en forma directa con la interfaz y el desempeño del usuario eficaz y eficientemente.

La propuesta, entonces, busca enfrentar el desarrollo del producto de software desde una perspectiva centrada en los principios de usabilidad para lograr un producto de software de calidad en el uso. La tabla 1 muestra que cada tarea de usabilidad está asociada con uno o varios artefactos entregables con los que se quiere verificar su cumplimiento, p. ej. en la etapa de ingeniería de requisitos esta tabla muestra que se debe hacer la revisión de escenarios actuales,

la descripción de las características de usabilidad deseadas, las funcionalidades del software que tendrán mayor impacto en los usuarios y, por último, la definición de requisitos de uso y validación. Estas tareas se verifican con algunos artefactos entregables, como el modelado para la especificación de contexto de uso (MEC). El MEC es un artefacto que describe cuál será el entorno de uso de la aplicación, es decir, en qué condiciones y con qué herramientas se usará. También se debe entregar en esta etapa la lista de características de usabilidad (LCU), el documento de requisitos validados (DRV) y el documento de priorización de requisitos (DPR). A continuación se muestra la descripción completa de las tareas y artefactos de usabilidad que se deberán entregar en cada una de las etapas de desarrollo de software. Para claridad de las figuras posteriores, a cada artefacto se le asocia un identificador.

La figura 1 describe los actores involucrados en las actividades de usabilidad planteadas dentro de cada etapa de un proceso de desarrollo de software, y se especifican sus responsabilidades con respecto a los artefactos propuestos.

Para la etapa de *Ingeniería de Requisitos*, el desarrollador es responsable de los entregables planteados, pero necesita de los usuarios del sistema (avanzado e intermedio) para tomar en cuenta la visión de ellos sobre las tareas por desarrollar y sus expectativas sobre la interfaz. Para la etapa de *Análisis y Diseño*, los expertos en usabilidad (desarrollador y auditor) construyen las plantillas de listas de chequeo de usabilidad. El desarrollador debe plasmar el diseño de las interfaces de usuario y la especificación funcional.

Para la etapa de *codificación y pruebas*, los desarrolladores codifican teniendo en cuenta los requisitos de usabilidad establecidos y para cada versión hacen evaluaciones de usabilidad con las plantillas que ellos mismos construyeron. También se encargan de la depuración de la interfaz. Si se utilizan ingenieros de pruebas, serían estos quienes



Tabla 1. Tareas y artefactos de usabilidad propuestos

Tareas de usabilidad	Artefactos entregables
Ingeniería de Requisitos (<i>análisis del negocio, especificación de requisitos funcionales y no funcionales, validación de requisitos</i>)	
Revisión de escenarios actuales o deseados (usuarios, tareas, ambientes) Descripción de características de usabilidad deseadas Definición de las funciones del software con mayor impacto en las tareas de los usuarios Definición de requisitos de uso y su validación	Modelado para la especificación del contexto de uso (MEC). Lista de características de usabilidad para tener en cuenta dentro del desarrollo (LCU). Documento de requisitos validado teniendo en cuenta aspectos de usabilidad (DRV) Documento de priorización de requisitos de usabilidad del producto (DPR)
Análisis y Diseño (<i>subsistemas, componentes, módulos de software, diseño de pruebas</i>)	
Diseño de escenarios y tareas de los usuarios (diseño de interacción) Diseño de la ayuda Diseño de listas de chequeo de usabilidad (plantillas)	Especificación funcional del producto con énfasis en el diseño detallado de la interacción (EF) Diseño de la interfaz de usuario (DIU) Plantillas de las listas de chequeo (PLC)
Codificación y Pruebas (<i>código fuente, ejecución de pruebas de los desarrolladores</i>)	
Implementación de las interfaces de usuario Validación de los prototipos del software contra la especificación, utilizando las listas de chequeo y la lista de características de usabilidad para tener en cuenta	Reporte de la evaluación de usabilidad realizada por los desarrolladores (REUD) Diagnósticos de los defectos de usabilidad (DDU) Lista de inconformidades para ser depuradas (LID) Interfaz de usuario validada y depurada (IUV)
Pruebas con Usuarios (<i>ejecución de pruebas α y β con usuarios seleccionados</i>)	
Validación de funciones teniendo en cuenta aspectos de usabilidad utilizando listas de chequeo	Resultados de las pruebas de usabilidad realizadas por los usuarios (RPUU) Diagnósticos de los defectos de usabilidad (DDU) Lista de inconformidades para ser depuradas (LI) Interfaz de usuario depurada nuevamente (IUV)
Verificación (<i>verificación funcional, verificación del producto, verificación del sistema</i>)	
Verificación de funciones teniendo en cuenta aspectos de usabilidad empleando listas de chequeo	Resultados de la evaluación de usabilidad realizada por los auditores del sistema (REUA) Diagnósticos de los defectos de usabilidad (DDU) Lista de nuevos requisitos de usabilidad e inconformidades (LI)

realicen las evaluaciones heurísticas de usabilidad junto con los desarrolladores. Para la etapa de *pruebas con usuarios*, los usuarios ejecutan las pruebas de usabilidad basándose en las plantillas de listas de chequeo *construidas por los desarrolladores o auditores*; los desarrolladores deben hacer el análisis de los resultados y proponer las listas de depuración del software. Finalmente, para la etapa de *verificación*,

los auditores realizan las evaluaciones de usabilidad con las plantillas de *listas de chequeo que ellos mismos construyeron*, deben realizar también el análisis de problemas encontrados y sintetizar una lista de inconformidades o nuevos requisitos. Es importante aclarar que cada etapa genera una serie de artefactos que son insumo para la etapa siguiente, como se aprecia en la figura 1; el proceso se concibe

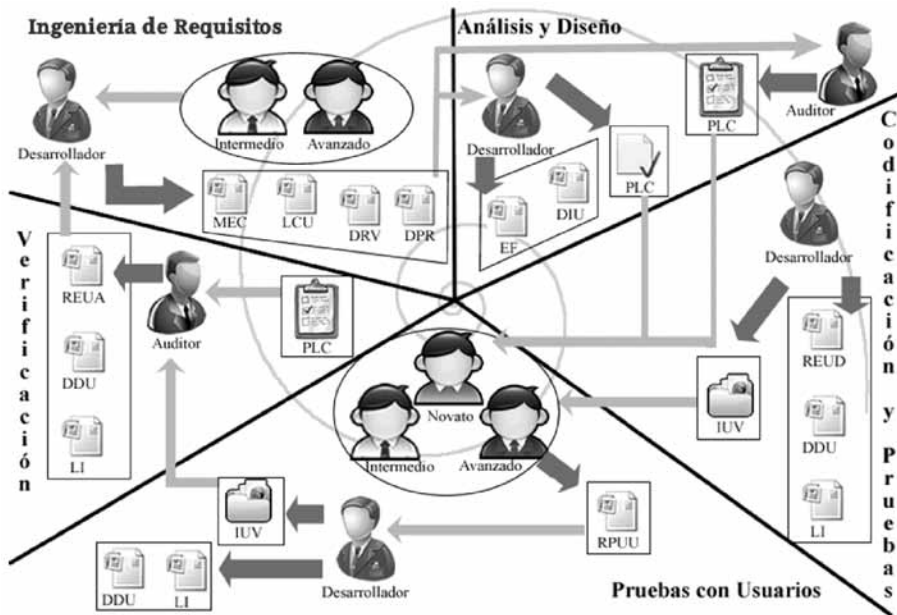


Figura 1. Actores involucrados en el desarrollo de las tareas de usabilidad planteadas para el ciclo de vida de desarrollo

de manera cíclica e iterativa, en el cual de acuerdo con cada ciclo se van depurando gradualmente las características de usabilidad de la aplicación.

Las listas de chequeo clásicas encontradas hasta el momento califican las características de usabilidad como “cumple” o “no cumple”. De esta manera, se pueden obtener resultados imprecisos, ya que es probable que la característica evaluada se encuentre presente pero no está totalmente implementada. En este caso no se debería calificar “cumple” o “no cumple”, sino que sería mejor expresarlo como una expresión numérica de valores de verdad sobre la totalidad del cumplimiento. Se propone modificar las listas de chequeo clásicas, que en adelante denominaremos plantillas, agregando para cada característica que se evalúa tres metadatos sobre la percepción del evaluador; según se enuncian a continuación.

Ponderación de importancia de cada característica para el evaluador. Recopilar la impresión del evaluador sobre la relevancia que tiene en él la

característica permitirá más adelante clasificar las características de usabilidad más relevantes para cada tipo de usuario evaluador. La ponderación de importancia se califica entre 1 (poco importante) y 5 (fundamental).

Calificación del nivel de cumplimiento de cada característica para el evaluador. El grupo investigador decidió que en muchos casos el usuario puede juzgar que la característica se cumple en algún nivel. Por esto, se decidió incluir la calificación como la expresión de un nivel de cumplimiento entre 0 (la característica no se cumple en absoluto) y 100 (la característica se cumple por completo). Esto permite definir el valor porcentual de conformidad sobre cada característica evaluada desde la perspectiva de cada evaluador de la interfaz.

Justificación de la calificación. Con este metadato se indaga sobre las causas que pueden tener las calificaciones altas y bajas de nivel de cumplimiento y las razones por las cuales los usuarios ponderan como alta o baja una característica de usabilidad.



Para evitar la dificultad de llegar a un consenso sobre las justificaciones lo mejor es proporcionar una lista fija de posibles justificaciones.

La tabla 2 describe el esquema general de las listas de chequeo modificadas. En la sección 4 se mostrarán algunos ejemplos de las plantillas construidas para la evaluación del software PREDICA.

4. APLICACIÓN DE LA PROPUESTA

El producto de software particular, en cuyo desarrollo se aplicó la propuesta, es una biblioteca digital web de consulta de documentos en el área de las ciencias de la computación (PREDICA). El análisis de la primera versión del software, hecho con la técnica de foro de discusión dirigido, permitió evolucionar conceptualmente la interfaz hasta el punto de sustentar la utilización de aplicaciones ricas en internet (RIA, por su sigla en inglés de *Rich Internet Application*) como un nuevo paradigma de interacción para las versiones siguientes. Utilizar los conceptos de RIA implica enfocarse en nuevas herramientas que enriquecen la interacción del usuario, amplían su experiencia y relación con la aplicación (Eichorn, 2006; O'Reilly, 2005).

La figura 2 muestra las perspectivas de usuarios, entregables y tareas para lograr la evaluación de usabilidad en ciclo de desarrollo empleado para la aplicación PREDICA.

Se agruparon por lo menos 12 usuarios finales de pruebas en cada uno de los cinco grupos propuestos, de acuerdo con su afinidad a Internet y con el porcentaje de actividades dedicadas a la consulta de material bajo este ambiente. Adicionalmente, se consideraron los aspectos culturales (nacionalidad, región, formas de expresión) y del entorno circundante al usuario (aplicaciones bajo ambiente web con las cuales está familiarizado el usuario). Para este proyecto no se tuvieron en cuenta aspectos étnicos ni de sexo, pues se consideraron irrelevantes para la evaluación propuesta.

Se desarrollaron dos grupos de plantillas o archivos diferentes con listas de chequeo, partiendo de las listas de chequeo básicas. Estas plantillas fueron realizadas por los dos grupos de expertos (desarrolladores y auditores del sistema). La primera plantilla de cada grupo evalúa la usabilidad de los formularios de consulta en la interfaz de consulta general y en la avanzada. La segunda plantilla de cada grupo evalúa la página de resultados de los documentos recuperados del módulo de consultas generales y avanzadas. La tabla 3 muestra un resumen de una lista de chequeo elaborada por los desarrolladores para evaluar la usabilidad de los formularios de consulta. La tabla 4 muestra un resumen de una lista de chequeo elaborada por los desarrolladores para evaluar la usabilidad de la página de resultados. La tabla 5 recopila las características de las plantillas de listas de chequeo.

Tabla 2. Esquema general de las listas de chequeo modificadas

Característica de usabilidad por comprobar	Ponderación de importancia 1 (poco), 5 (fundamental)	Nivel de cumplimiento (0-100)	Justificación
Pregunta que indaga sobre el cumplimiento de una característica de usabilidad	—	—	<input type="checkbox"/> Opción 1 ... <input type="checkbox"/> Opción N

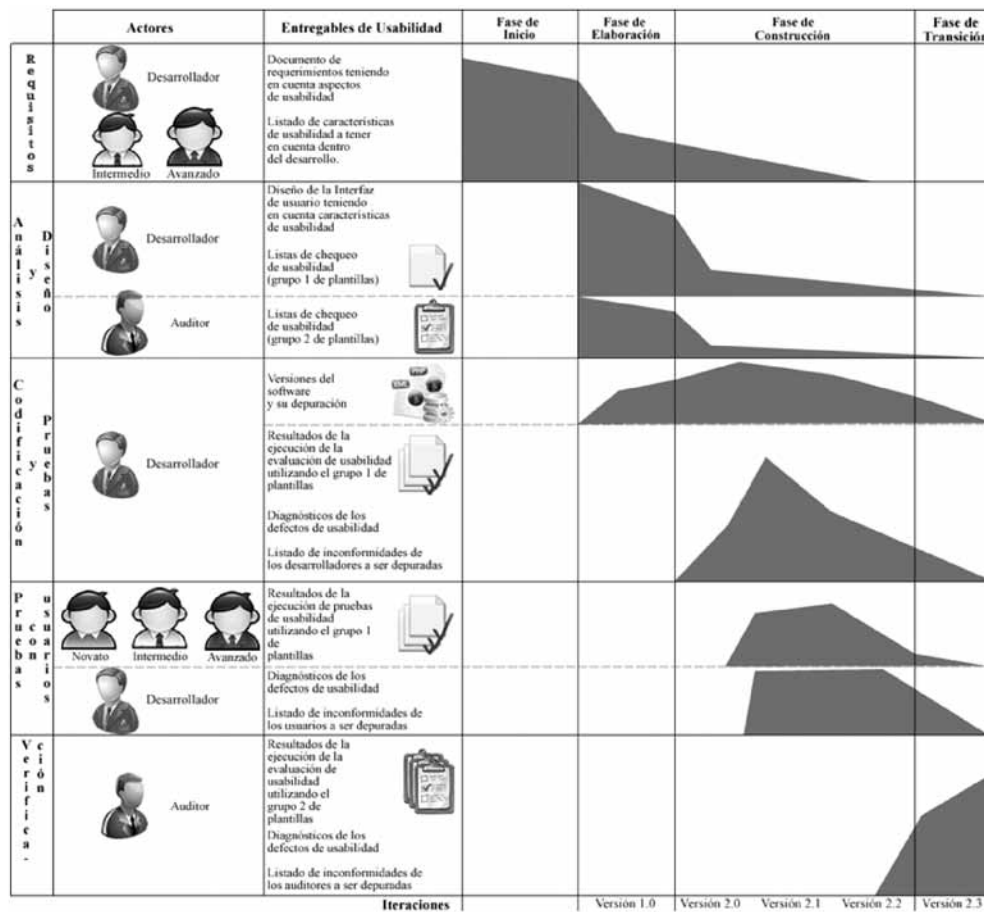


Figura 2. Perspectivas de usuarios, entregables y tareas para lograr la evaluación de usabilidad en ciclo de desarrollo empleado para la aplicación PREDICA

Tabla 3. Resumen de lista de chequeo por los desarrolladores sobre formularios de consulta

Características de usabilidad	Nivel de importancia (1-5)	Cumplimiento (0 % - 100 %)	Justificación
¿Al entrar a la aplicación se despliega automáticamente la interfaz de consulta general?			
¿La interfaz de consulta general contiene un cuadro de texto para introducir los términos de la consulta?			
¿La interfaz de consulta general contiene un botón con la cadena “buscar” y no otra, para activar la consulta?			
¿El área de búsqueda está identificada con un encabezado que titula la opción de búsqueda?			
¿El cuadro de texto soporta una cantidad de caracteres adecuada para que el usuario pueda escribir la consulta?			



Tabla 4. Resumen de lista de chequeo por los desarrolladores sobre página de resultados

Características de usabilidad	Nivel de importancia (1-5)	Cumplimiento (0 % - 100 %)	Justificación
¿Los resultados se muestran en la misma interfaz de consulta, de tal forma que el usuario no pierde el enfoque general de la aplicación?			
¿Los resultados se muestran de forma lineal, de tal forma que los primeros son aquellos que tienen mayor relevancia con respecto a todos los términos especificados en la consulta?			
¿Cada resultado tiene un enlace visible para que el usuario acceda al documento buscado o a visualizar más información de éste?			
¿En la página de resultados aparece claramente un resumen o palabras clave que le muestran al usuario el tema de cada documento encontrado?			
¿La interfaz le muestra al usuario el número de documentos recuperados?			
¿La página de resultados mantiene visible la consulta que el usuario hizo para que pueda ver lo que se encontró con respecto a lo buscado?			

Tabla 5. Características de las plantillas de listas de chequeo

Grupo de expertos	No. de características recopiladas	No. de plantillas construidas	Interfaces que se someten a pruebas de usabilidad
Desarrolladores (5 personas)	50	2	Formulario de consulta general y de consulta avanzada. Resultados de consulta general y de consulta avanzada
Auditores (12 personas)	65	2	

Los resultados de las primeras evaluaciones heurísticas elaboradas por los desarrolladores y la depuración subsecuente del software contribuyeron a la construcción de la versión 2.0 de la interfaz (figura 3).

En las pruebas de usabilidad, los usuarios no conocían de antemano la aplicación. Al grupo de usuarios de cada grupo se les proporcionaron las plantillas de evaluación y se les invitó a utilizar la

aplicación sin ningún manual ni ayuda. Los usuarios reportaron su experiencia con PREDICA calificando cada una de las características de usabilidad en las plantillas.

El análisis de los resultados de las pruebas con usuarios y las evaluaciones heurísticas de los desarrolladores llevaron a las depuraciones de la interfaz para producir las versiones 2.1 y 2.2 de la interfaz (figura 4).

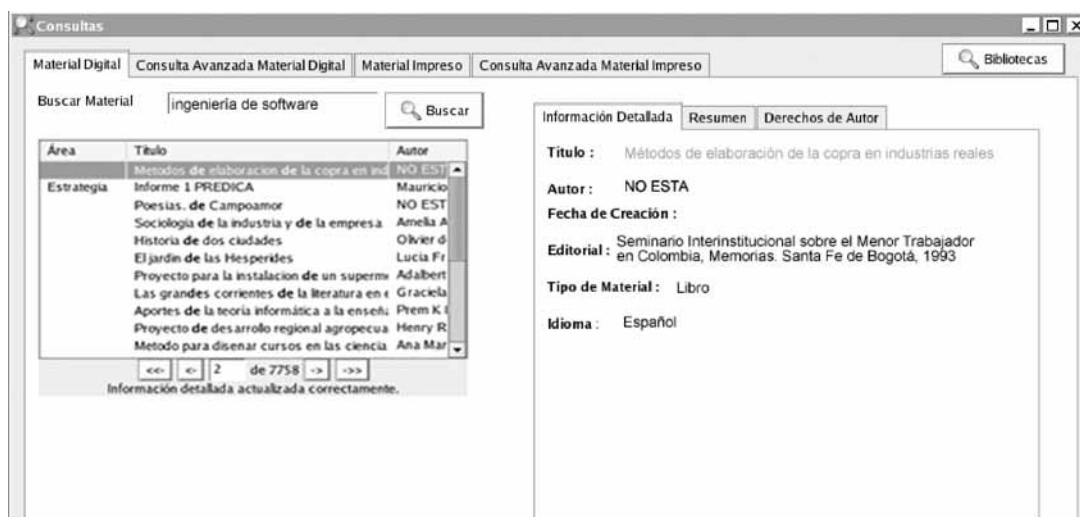


Figura 3. Versión 2.0 de la interfaz de consulta general de PREDICA

Por último, las evaluaciones heurísticas realizadas por el grupo de auditores del sistema con sus propias plantillas, cuyos resúmenes se presenta en la

tabla 6 y la tabla 7, generaron una tercera ronda de depuraciones con la que se llegó a la versión actual 2.3 de la interfaz (figuras 5 y 6).



Figura 4. Versión 2.2 de la interfaz de consulta general de PREDICA



Tabla 6. Resumen de lista de chequeo por los auditores para formularios de consulta

Características de usabilidad	Nivel de importancia (1-5)	Porcentaje de cumplimiento (0 % - 100 %)	Justificación
¿El usuario tiene cómo elegir el intercambio entre búsqueda general y avanzada?			
¿Las convenciones de navegación son consistentes en todo el sitio web?			
¿El tamaño de la letra es lo suficientemente grande para todos los usuarios?			
¿Se evita generar ventanas sobre ventanas para visualizar los detalles de un elemento?			

Tabla 7. Resumen de lista de chequeo por los auditores para evaluar página de resultados

Características de usabilidad	Nivel de importancia (1-5)	Cumplimiento (0 % - 100 %)	Justificación
¿Se presentan los resultados como una lista de resultados, en orden descendente, según alguna estimación de relevancia?			
¿Cada resultado tiene un enlace que permita encontrar más información?			
¿Los resultados tienen un resumen de lo encontrado?			
¿Cada resultado tiene un título descriptivo?			
¿Se categorizan los resultados, de tal manera que los mejores sean los primeros?			

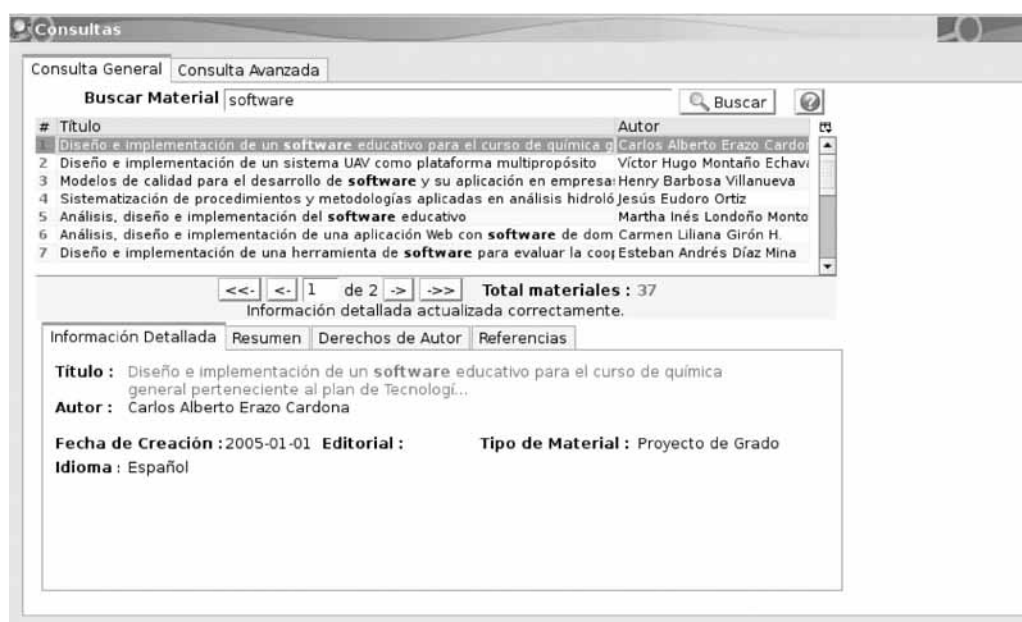


Figura 5. Versión 2.3 de la interfaz de consulta general de PREDICA

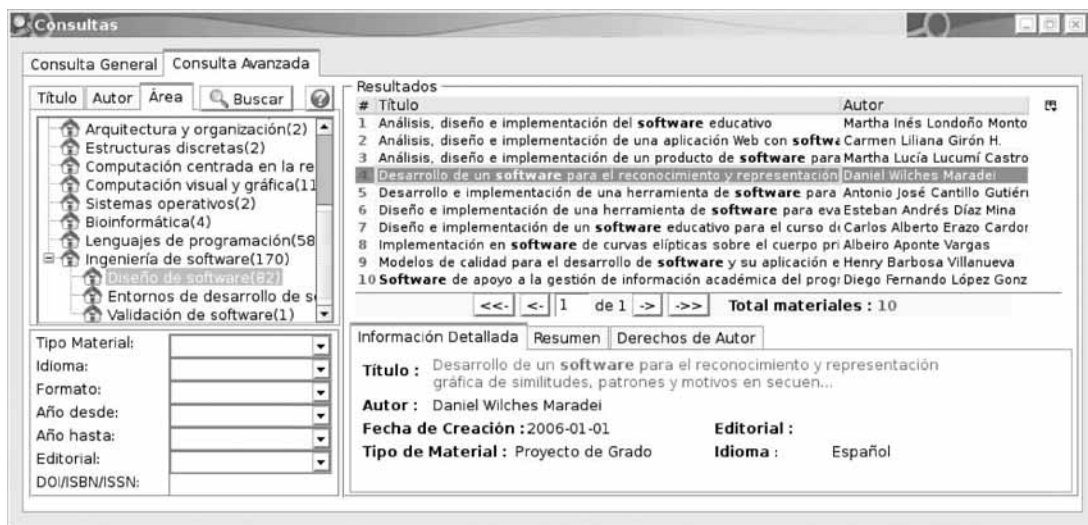


Figura 6. Versión 2.3 de la interfaz de consulta avanzada por áreas de conocimiento

5. RESULTADOS Y DISCUSIÓN

El grupo de investigadores deseaba indagar si la valoración sobre el cumplimiento de las características de usabilidad en el software es subjetiva al tipo de usuario que realiza la ejecución de las pruebas o evaluaciones. Por tanto, las plantillas propuestas para las listas de chequeo formulan una calificación porcentual y no una calificación absoluta (SÍ o NO) sobre el cumplimiento de cada característica de usabilidad listada (ver tablas 3-6). Para el caso de estudio realizado con PREDICA, el grupo de investigadores realizó el análisis de las calificaciones recopiladas que hicieron los desarrolladores y los tres grupos de usuarios. En este análisis, para el campo cuantitativo “*Nivel de cumplimiento*” muestra una tendencia que indica que, según el tipo de usuario, la calificación promedio era diferente. La figura 7 permite el nivel de satisfacción de los diferentes tipos de usuario frente a la usabilidad del sistema para la primera ronda de pruebas conjunta sobre la versión 2.0 del sistema PREDICA. Es importante destacar que en este contexto la satisfacción no se refiere a la apreciación estética y satisfacción subjetiva de los

elementos gráficos de la interfaz de usuario, sino a la satisfacción de usuario respecto al cumplimiento o no de las características de usabilidad presentadas en la plantilla.

Para las pruebas sucesivas con usuarios, la calificación del campo cuantitativo “*Nivel de cumplimiento*” muestra una tendencia que indica que el nivel de satisfacción general sobre la usabilidad del software aumentaba tras cada depuración. La figura 8 permite apreciar el aumento en el nivel de satisfacción de los diferentes tipos de usuario frente a la usabilidad del sistema para cada una de las versiones del software.

Con el análisis del campo cuantitativo de “*Ponderación de importancia*” el grupo de investigadores también pudo establecer como tendencia que la calificación del campo para cada característica evaluada en las listas de chequeo es diferente según el tipo de usuario. La figura 9 muestra el grado de ponderación de importancia general de las características de usabilidad de acuerdo con el tipo de usuario para la versión 2.0 de la interfaz.

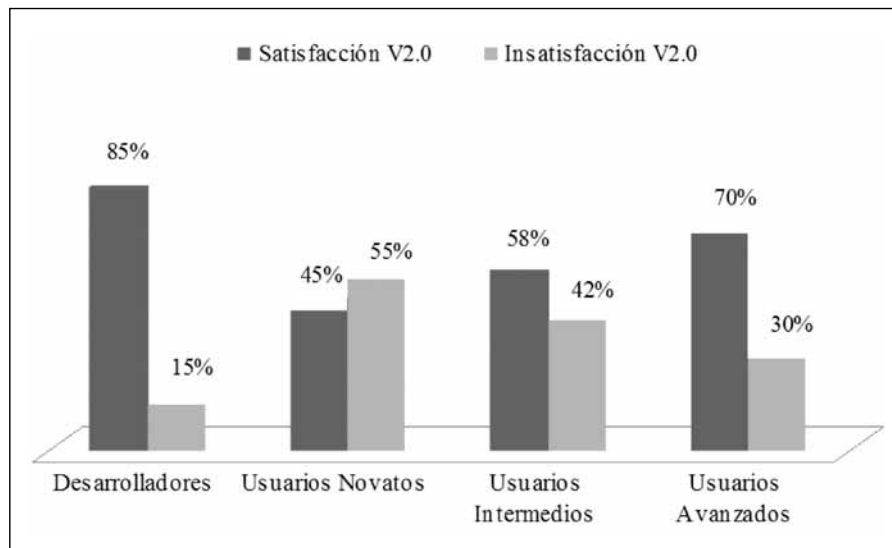


Figura 7. Calificaciones del cumplimiento de usabilidad general para la versión 2.0

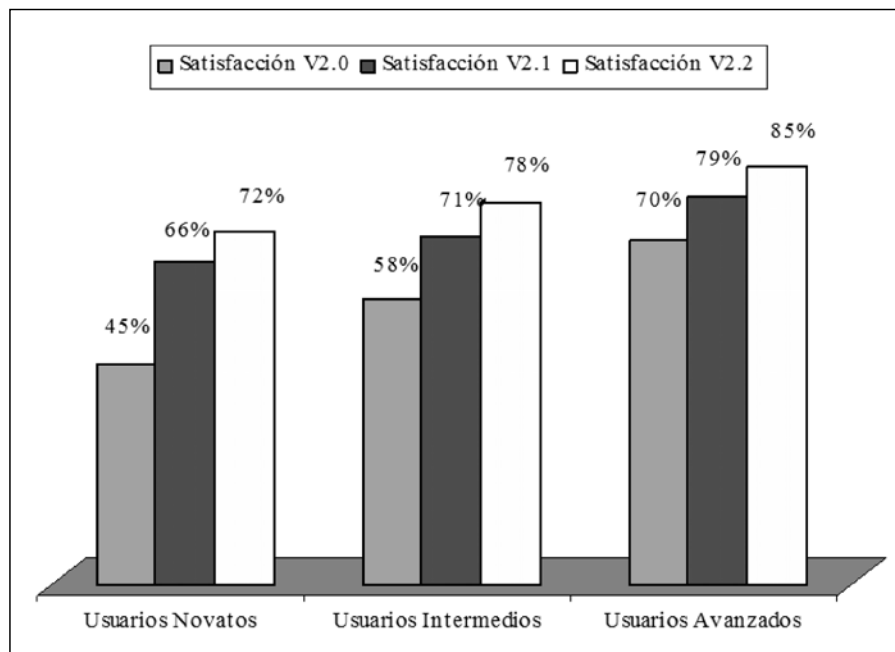


Figura 8. Calificaciones de los usuarios del cumplimiento de usabilidad general para cada versión

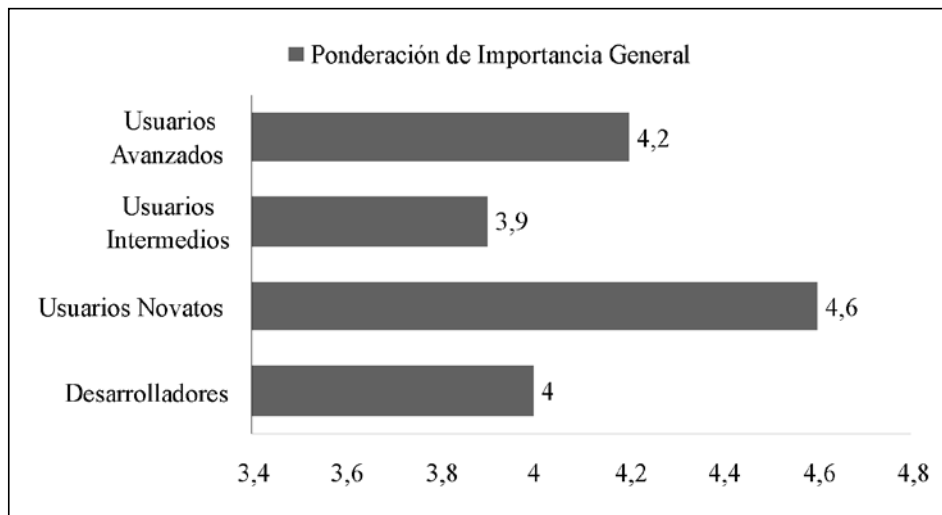


Figura 9. Ponderación de importancia general de usabilidad para la versión 2.0

Con la investigación también se evidenció la tendencia de que las listas de chequeo construidas por auditores son más exhaustivas que las construidas por los desarrolladores. En el caso de PREDICA, el análisis sobre el número de características de usabilidad presentes en las listas de chequeo que muestra la tabla 5 permite observar que el número de características recopiladas por los auditores superan en un 30 % al número de características recopiladas por los desarrolladores.

6. CONCLUSIONES

Esta propuesta de actividades y artefactos es un ejemplo en la literatura que apoya la idea de que introducir perspectivas de usabilidad desde etapas tempranas del desarrollo de software permite alcanzar un mejor nivel de depuración de la interfaz antes de emplear la aplicación de software. Utilizando esta propuesta, se introducen conceptos de calidad de interfaces de usuario durante el proceso de desarrollo de software, garantizando la usabilidad de los usuarios al final de la entrega del producto.

Al utilizar la combinación de diferentes técnicas de calificación de usabilidad se potencian las recomendaciones para la depuración de la interfaz de usuario. Las diferentes técnicas permiten evaluar de manera separada la usabilidad encontrando algunas recomendaciones comunes y otras propias. Luego se pueden confrontar resultados para establecer de forma rápida prioridades sobre las recomendaciones que serán depuradas inicialmente y el orden de depuración para las subsecuentes.

La reutilización de las plantillas de listas de chequeo mejoradas, planteada en esta propuesta, permite realizar la evaluación heurística de los expertos en usabilidad y también ser utilizadas como guía para el desarrollo de las pruebas con usuarios. Esta reutilización permite la aplicación de estas dos técnicas con ahorro de tiempo y dinero, ya que se suprime la elaboración de diferentes tipos de artefactos para ambas técnicas. Adicionalmente, la nueva manera propuesta de calificar las listas de chequeo brinda más herramientas de información sobre la percepción de los diferentes tipos de usuarios y sus razones. El campo cualitativo “Justificación”, agregado a las listas de chequeo, permite indagar las razones de la calificación de los usuarios para cada característica,



muy útil sobre todo para aquellas calificadas con bajo nivel de cumplimiento que generan elementos concretos para depurar la interfaz. Para evitar un trabajo tedioso al analizar diferentes justificaciones y hallar opiniones comunes entre los usuarios, se recomienda ofrecer una lista de justificaciones predefinidas para cada característica evaluada. En la construcción de las listas de chequeo deben tenerse en cuenta los patrones ya definidos de modelos de interfaces web y comportamientos esperados por los usuarios e incluso también los comportamientos no deseados (antipatrones). Estos patrones y antipatrones de usabilidad no deben ser ignorados y, por el contrario, es un éxito tenerlos en cuenta y procurar su evaluación desde etapas tempranas del desarrollo.

La propuesta de clasificación de usuarios presentada es adaptable a cualquier tipo de desarrollo. La idea propuesta de utilizar auditores como evaluadores en usabilidad es una técnica que les permite realizar su tarea de inspección de manera sistemática y correcta frente al sistema construido, evaluando no solo la funcionalidad, sino la interacción de los usuarios; cabe resaltar, como se enunció en la sección 2, que el auditor no se encuentra inmiscuido en las etapas de análisis ni de la implementación de software, lo cual le da un carácter más general respecto a la evaluación que realiza; el acierto se percibe en la interacción con el grupo de desarrollo y la manera como su perspectiva exógena contribuye a la realimentación y, por ende, a la correcta evaluación del proyecto. También hay que reconocer los beneficios de introducir las listas de chequeo elaboradas por desarrolladores desde etapas tempranas del proceso de desarrollo de software. Por tanto, es bueno utilizar estos dos tipos de expertos para preparar listas de chequeo, con el fin de ejecutar evaluaciones de usabilidad desde etapas tempranas y también para hacer una revisión más exhaustiva para las etapas finales del proceso de desarrollo. Se comprobó el beneficio planteado de emplear al desarrollador como un experto de usabilidad para la producción de listas de chequeo y además utilizarlo para la ejecución de estas evaluaciones. Pero dada la

diferencia en la ponderación de importancia y nivel de cumplimiento de las características de usabilidad entre los desarrolladores y los usuarios, es claro que no se pueden dejar de utilizar los usuarios para la ejecución de las pruebas de usabilidad.

La evaluación de la interfaz de PREDICA fue un buen ejercicio, ya que permitió, mediante un desarrollo experimental, comprobar el aumento de los principios de usabilidad y disminuir la brecha entre las necesidades y expectativas de los usuarios y su funcionalidad. Esto se evidencia en el aumento de la calificación del “*Nivel de cumplimiento*” sobre los aspectos de usabilidad en el software para las pruebas realizadas a los prototipos sucesivos de la interfaz.

7. TRABAJO FUTURO

Las tecnologías de aplicaciones web 2.0 proponen nuevos retos en el área de evaluación de usabilidad web. Considerando que hay un gran acercamiento en la web 2.0 hacia aplicaciones cada vez más parecidas a las aplicaciones de escritorio, esto disminuye la brecha entre la usabilidad web y la usabilidad de aplicaciones de escritorio, además, estas tecnologías van en contra de ciertos paradigmas de comportamiento de las aplicaciones web tradicionales y de patrones de usabilidad establecidos para el web tradicional. Por lo tanto, se deben realizar estudios de usabilidad para el web 2.0, con el objeto de definir nuevos patrones de usabilidad web y adaptar algunos que eran tradicionales en la web 1.0, pero que ya no son aplicables en aplicaciones RIA.

AGRADECIMIENTOS

El trabajo descrito en este artículo se enmarca en el proyecto de investigación PREDICA, llevado a cabo por la Escuela de Ingeniería de Sistemas y Computación (EISC) de la Universidad del Valle y financiado por el Instituto Colombiano para el Desarrollo de la Ciencia y la Tecnología “Francisco José de Caldas” (Colciencias). Se resalta la labor de los ingenieros de desarrollo de PREDICA que

colaboraron con las actividades de pruebas: Javier E. Carrillo y Mauricio Ciprián. También se resalta la labor de los estudiantes de la asignatura *Técnicas de Pruebas de Software, semestre 2008-A* de la Universidad del Valle, que desempeñaron el rol de auditores de software. Por último, los autores expresan su reconocimiento a las docentes Paola J. Rodríguez, Ivette Kafure y María Eugenia Valencia, quienes trabajaron con este equipo de investigación en las pruebas de usabilidad de PREDICA aplicando la técnica de análisis de la tarea y la actividad.

REFERENCIAS

- Alarcón, H. F.; Hurtado, A. M.; Pardo, C.; Collazos, C. A. y Pino, F. J. (2007). Integración de técnicas de usabilidad y accesibilidad en el proceso de desarrollo de software de las MiPyMEs. *Revista Avances en Sistemas e Informática*, vol. 4, No. 3, pp. 149-156.
- Aveledo, M. and Moreno, A. M. (2008). "Responsibilities in the usability requirements elicitation process". *Journal of Systemics, Cybernetics and Informatics*, vol. 6, No. 6, pp. 54-60.
- Aveledo, M.; De la Rosa, A. and Moreno, A. *Usability design recommendations: a first advance*, in CISSE 2008. Disponible en <http://conference.cisse2008.org/schedule.aspx>.
- Aveledo, M. and De la Rosa, A. "Incorporating usability in the software development process" in the Proceedings of International Association of Science and Technology for Development 2010 (IASTED 2010) on Software Engineering.
- Bevan, Nigel. "Extending quality in use to provide a framework for usability measurement". Lecture Notes in Computer Science, Berlin/Heidelberg: Springer, vol. 5619/2009, Book Human Centered Design, 2009, pp. 13-22.
- Cheikhi, L.; Abran, A. and Suryn W. "Harmonization of usability measurements in ISO9126 software engineering standards". *IEEE ISIE*. 2006. Montreal, Quebec, Canada. pp. 3246-3251.
- Cysneiros, L. M. and Kushniruk, A. 2003, "Bringing usability to the early stages of software development". *Requirements Engineering Conference*. 2003. *Proceedings. 11th IEEE International*, pp. 359-360.
- Eichorn, J. *Understanding AJAX: Using JavaScript to create rich internet applications*. New Jersey: Prentice Hall, 2006. pp. 4-12.
- Ferré, X. *Incrementos de usabilidad al proceso de desarrollo software*. VIII Jornadas de Ingeniería del Software y Bases de Datos, 2003 JISBD. Alicante, España (12-14 noviembre, 2003). pp. 293-302.
- Florián, B. E.; Valencia, M. E.; Rodríguez, P. J.; Millán, M.; Gaona, C. M.; Carrillo J. E. y Ciprián M. (2007). "Diseño de una plataforma experimental para la búsqueda y recuperación de documentos en una biblioteca digital". *Ingeniería y Competitividad*, vol. 9, No. 2, pp.105-117.
- González, M. P.; Lorés, J. y Pascual, A. *Evaluación heurística*. Universidad de Lleida. 2001. Consultado el 23 de enero de 2008. Disponible en: <http://griho.udl.es/ipo/ipo/pdf/15-Evaluacion-Heuristica.pdf>
- Granollers, T.; Lorés, J.; Sendín, M. y Perdrix, F. (2005). Integración de la IPO y la ingeniería del software: MPlu+a. III. Taller en Sistemas Hipermedia Colaborativos y Adaptativos SIHICA'2005 (Granada, España, septiembre 13-16, 2005). Disponible en <http://www.aipo.es/items.php?id=83>.
- Granollers, T.; Lorés, J. y Cañas J. J. *Diseño de sistemas interactivos centrados en el usuario*. UOC. 2005. <http://griho.udl.es/mpiu/mpiu/index.htm> Consultado en octubre de 2007.
- Hakiel, S. "Delivering ease of use" (1997). *Computer & Control Engineering Journal*, vol. 8, No. 2 (Apr.), pp. 81-87.
- ISO/IEC 25010-3: *Systems and software engineering: software product quality and system quality in use models*. 2009. Available in: http://www.iso.org/iso/catalogue_detail.htm?csnumber=35733.
- Juristo, N.; Moreno, A. M. and Sanchez-Segura, M. I. (2007). "Guidelines for eliciting usability functionalities". *IEEE Transactions on Software Engineering*, vol. 33, No. 11, pp. 744-758.
- Kafure, I. *Validação do formalismo TAOS para a análise da tarefa no contexto da concepção de interfaces homem-computador*. Tesis (Maestría). Universidade Federal de Campina Grande. <http://copin.ufcg.edu.br/> 2000. Consultado en enero de 2008.
- Kafure, I. *Usabilidade da imagem na recuperação da informação no catálogo público de acesso em linha*. Tesis de Doctorado, Universidade de Brasília, Departamento de Ciência da Informação e Documentação. 2004.



19. Kafure, I.; Valencia, M. E.; Rodríguez, P. J.; Florián, B. E.; Carrillo J. E.; Ciprián, M. y Solarte O. *Evaluación de la usabilidad de la biblioteca digital PREDICA*. Memorias del Seminario Internacional de Bibliotecas Digitales, Brasil. 2007. Disponible en <http://libdigi.unicamp.br/document/?code=23485>.
20. LabiUtil - Laboratório de Utilizabilidade da Informática. *Critérios ergonômicos*. <http://www.labiutil.inf.ufsc.br/CriteriosErgonomicos/LabiUtil2003-Crit/100conduc.html>. 2007. Consultado en septiembre de 2007.
21. Mayhew, D. *The usability engineering lifecycle*. Morgan Kaufmann, 1999. pp. 339-483.
22. Medeiros, J. H.; Kafure, I. and Lula, B. "TAOS a task-and-action oriented framework for user's task analysis in the context of computer interfaces design". *IEEE Computer Society Press*. 2000. SCCC '00. Proceedings. XX International Conference, 16-18 Nov. 2000, pp. 24-31.
23. Nielsen, J. *Usability engineering*. Academic Press, 1993. pp. 115-206.
24. Nielsen, J. *Designing web usability: the practice of simplicity*. Indianapolis: New Riders Publishing, 1999. pp. 129-155, 224-238.
25. Nielsen, J. and Loranger, H. *Prioritizing web usability*. Berkeley: New Riders Press, 2006. pp. 57-121, 137-170.
26. O'Reilly, T. What is web 2.0: *Design patterns and business models for the next generation of software*. [online] 2005. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>. (2005). Consulted on January 23, 2008.
27. Ramli, R. and Jaafar, A. e-RUE: A cheap possible solution for usability evaluation, *Information Technology, 2008. ITSIM 2008. International Symposium on Informatic Technology*, vol. 3, 2008. pp.1-5.
28. Rosenbaum, S. "Usability evaluations versus usability testing: when and why? Professional Communication". *IEEE Transactions*, vol. 32. 1989. pp. 210-216.
29. Rubin, J. and Chisnell, D. *Handbook of usability testing: how to plan, design, and conduct effective tests*. Wiley Technical Communications, 2008. 2nd ed., chapter 1.
30. Shneiderman, B. y Plaisant, C. *Diseño de interfaces de usuario*. Madrid: Pearson Education. 2006. pp. 143-145.
31. Singh, M. *U-SCRUM: An agile methodology for promoting usability*. In: *AGILE '08*, Toronto, 2008. pp. 555-560
32. Tao, Y. "Introducing usability concepts in early phases of software development". In: *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*. Indianapolis, 19-22 Oct. 2005. pp. T4C - 7-8.
33. Tidwell, J. *Designing interfaces: patterns for effective interaction design*. O'Reilly, 2006. pp. 207-239.