

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/221595210>

Principios Básicos de Usabilidad para Ingenieros Software.

Conference Paper · January 2000

Source: DBLP

CITATIONS

6

READS

1,289

1 author:



[Xavier Ferre](#)

Universidad Politécnica de Madrid

30 PUBLICATIONS 523 CITATIONS

SEE PROFILE

All content following this page was uploaded by [Xavier Ferre](#) on 11 June 2014.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Principios Básicos de Usabilidad para Ingenieros Software

Xavier Ferré Grau

Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo
28660 – Boadilla del Monte (Madrid)
xavier@fi.upm.es

Resumen.

La usabilidad es un tema que está cobrando una importancia cada vez mayor en el desarrollo de software. A pesar de ello, la Ingeniería del Software sigue centrándose casi exclusivamente en atributos del software más relacionados con el interior del sistema, como el rendimiento o la fiabilidad. En el entorno actual, en el que los sistemas software están dirigidos a un público cada vez más amplio, a usuarios cada vez menos expertos en el manejo de sistemas informáticos, la usabilidad está destacándose como atributo fundamental para el éxito de un producto software. En este trabajo se muestran los principios básicos de la Ingeniería de Usabilidad junto con las principales técnicas que emplea, para una audiencia formada por profesionales de la Ingeniería del Software.

Palabras Clave: Ingeniería de Usabilidad, usabilidad, Ingeniería del Software, Ingeniería de Requisitos, análisis de tareas, test de usabilidad.

Introducción

En el concepto tradicional de calidad de un sistema software, la usabilidad se presenta como un atributo del software al que se presta especial atención en determinados proyectos software, de forma similar a como se trata la seguridad, por ejemplo. Este enfoque provoca que un gran número de sistemas software construidos tengan un nivel de usabilidad deficiente, cuando un mayor cuidado por este tema hubiera producido un sistema de mayor calidad percibida por el cliente sin un aumento excesivo en el coste de desarrollo.

Sin embargo, la creciente demanda de software más usable está cambiando este panorama. En las versiones más recientes de los estándares de ISO sobre calidad del software, ya se puede observar cómo la calidad de un sistema se distingue entre calidad inherente del software y calidad de uso [ISO98a] e [ISO00].

En este trabajo se muestra una visión general de la Ingeniería de la Usabilidad y su cometido en el desarrollo de sistemas software. Se comienza describiendo los atributos que forman la usabilidad de un sistema software, para delimitar a continuación el concepto de usabilidad en relación tanto con la interfaz gráfica de usuario como con el ciclo de vida software. Así mismo se justificará la necesidad de preocuparse por la usabilidad en el desarrollo de software. Seguidamente se introducirán las principales técnicas que nos permiten desarrollar un sistema más usable y asegurar que se ha alcanzado el nivel de usabilidad requerido. Estas técnicas son presentadas según su papel en el ciclo iterativo de desarrollo diseño-evaluación-rediseño, con una definición previa de los niveles de usabilidad que se pretende alcanzar. Una vez descritas las técnicas se detalla cuáles podrían ser los primeros pasos para implantar dichas técnicas en una organización de desarrollo de software. Para terminar se presentan los temas a resolver de cara a una posible integración de la Ingeniería de Usabilidad con la Ingeniería del Software.

Delimitación del Concepto de Usabilidad

Usabilidad se define en el estándar ISO 9241 como “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” [ISO98b], y en el estándar ISO 14598-1 [ISO98a] se define calidad de uso de forma análoga. Como se puede comprobar, en esta definición se liga la usabilidad de un sistema a usuarios, necesidades y condiciones específicas. Por tanto, la usabilidad del sistema no es un atributo inherente al software, no puede especificarse independientemente del entorno de uso y de los usuarios concretos que vayan a utilizar el sistema.

La Ingeniería de Usabilidad se puede definir como una aproximación al desarrollo de sistemas en la que se especifican niveles cuantitativos de usabilidad a priori, y el sistema se construye para alcanzar dichos niveles,

que se conocen como métricas [Preece94]. La Ingeniería de Usabilidad proporciona un modo práctico de asegurar que el software desarrollado alcanza un cierto nivel de usabilidad y está basada en la evaluación mediante test de usabilidad con usuarios.

La usabilidad no puede definirse como un atributo simple de un sistema, pues implicará aspectos distintos dependiendo del tipo de sistema a construir. Por ejemplo, un sistema para ser colocado en una pantalla táctil de la sala de un museo, debe llevar un software que requiera un entrenamiento mínimo con el sistema por parte del usuario, debido a que será utilizado por la mayoría de usuarios una sola vez en toda su vida. Aspectos de usabilidad como la eficiencia (número de tareas que se pueden realizar por hora) no son relevantes para este tipo de sistemas, mientras que la facilidad de aprendizaje es vital para el éxito del sistema. Por otra parte, el software que utiliza una teleoperadora para recoger los mensajes a enviar a un buscapersonas, puede requerir de un periodo de formación apreciable, pero interesa fundamentalmente que pueda realizar cada operación en el menor tiempo posible (eficiencia), para reducir el tiempo de espera de los clientes que utilizan el servicio.

Estos distintos aspectos de la usabilidad se denominan atributos de usabilidad.

Atributos de Usabilidad

La usabilidad es una cualidad demasiado abstracta como para ser medida directamente. Para poder estudiarla se descompone habitualmente en los siguientes cinco atributos básicos [Nielsen93]:

- **Facilidad de aprendizaje:** Cuán fácil es aprender la funcionalidad básica del sistema, como para ser capaz de realizar correctamente la tarea que desea realizar el usuario. Se mide normalmente por el tiempo empleado con el sistema hasta ser capaz de realizar ciertas tareas en menos de un tiempo dado (el tiempo empleado habitualmente por los usuarios expertos). Este atributo es muy importante para usuarios noveles.
- **Eficiencia:** El número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Lo que se busca es la máxima velocidad de realización de tareas del usuario. Cuanto mayor es la usabilidad de un sistema, más rápido es el usuario al utilizarlo, y el trabajo se realiza con mayor rapidez. Nótese que eficiencia del software en cuanto su velocidad de proceso no implica necesariamente eficiencia del usuario en el sentido en el que aquí se ha descrito.
- **Recuerdo en el tiempo:** Para usuarios intermitentes (que no utilizan el sistema regularmente) es vital ser capaces de usar el sistema sin tener que aprender cómo funciona partiendo de cero cada vez. Este atributo refleja el recuerdo acerca de cómo funciona el sistema que mantiene el usuario, cuando vuelve a utilizarlo tras un periodo de no utilización.
- **Tasa de errores:** Este atributo contribuye de forma negativa a la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea. Un buen nivel de usabilidad implica una tasa de errores baja. Los errores reducen la eficiencia y satisfacción del usuario, y pueden verse como un fracaso en la transmisión al usuario del modo de hacer las cosas con el sistema.
- **Satisfacción:** Éste es el atributo más subjetivo. Muestra la impresión subjetiva que el usuario obtiene del sistema.

Algunos de estos atributos no contribuyen a la usabilidad del sistema en la misma dirección, pudiendo ocurrir que el aumento de uno de ellos tenga como efecto la disminución de otro. Por ejemplo, esto puede ocurrir con la facilidad de aprendizaje y la eficiencia. Es preciso realizar el diseño del sistema cuidadosamente si se desea tanto una alta facilidad de aprendizaje como una alta eficiencia; siendo el uso de aceleradores (combinaciones de teclas que ejecutan operaciones de uso habitual) la solución más común para conjugar ambos atributos de usabilidad.

La usabilidad del sistema no es una simple adición del valor de estos atributos, sino que se define para cada sistema como un nivel a alcanzar para algunos de ellos.

Estos cinco atributos pueden descomponerse a su vez para conseguir una mayor precisión en los aspectos de usabilidad en los que se quiere poner mayor énfasis. Por ejemplo, *rendimiento en uso normal* y *uso de opciones avanzadas* son ambos subatributos de eficiencia, mientras que *primera impresión* es un subatributo de satisfacción.

Relación entre la Usabilidad y la Interfaz Gráfica de Usuario

En el desarrollo de software se identifica a menudo la usabilidad con las características de los elementos de una interfaz gráfica de usuario basada en ventanas, como puede ser su color, su disposición o el diseño gráfico de los iconos y animaciones. Sin embargo, la usabilidad no sólo tiene que ver con la interfaz gráfica de usuario.

La usabilidad de un sistema está ligada principalmente a la interacción del mismo, al modo en que se realizan las operaciones con el sistema. Esta interacción no está definida en la interfaz gráfica, sino que está imbricada en el código que implementa la funcionalidad del sistema. La interfaz gráfica de usuario es la parte visible de tal

interacción. Es cierto que la interfaz gráfica es una parte importante del sistema, y un buen diseño de la misma puede hacer que un sistema aumente su nivel de usabilidad, pero un sistema con un diseño de la interacción pobre no puede mejorar su nivel de usabilidad tan solo cambiando la interfaz gráfica.

En cuanto a usabilidad, la parte más crítica es la lógica del sistema (el concepto en base al cual funciona). Por tanto, la interacción debe diseñarse junto con la lógica de negocio, para asegurarnos de que la lógica del sistema es usable. No es posible diseñar la lógica de negocio independientemente de la interacción y luego unirlos. Esta relación entre diseño de la interacción y el diseño software tradicional obliga a modificar este último para acomodar al primero.

Por otra parte, la usabilidad se ocupa también de lo que es el entorno del sistema software propiamente dicho. Por ejemplo, se ocupa del sistema de ayuda, de la documentación de usuario, y del procedimiento de instalación.

Influencia en el Ciclo de Vida Software

Como se ha expuesto anteriormente, la usabilidad está en el interior del sistema, depende del concepto en el que se basa éste. Esta es la razón por la que es necesario ocuparse de la usabilidad del sistema desde el inicio de su desarrollo. No es razonable ocuparse de la usabilidad de un sistema cuando la mayor parte de los recursos dedicados a la construcción del mismo han sido ya consumidos. Por el contrario, cuando se pretende construir un sistema usable, es necesario pensar en el uso que éste va a tener desde el inicio, y hay que llevar a cabo evaluaciones de usabilidad desde las primeras etapas del proceso de desarrollo. Para tener algo que evaluar será necesario construir prototipos que den forma al diseño en el que se esté trabajando. El ciclo de vida usado deberá ser iterativo con prototipado.

Para construir un sistema que sea adecuado al uso que se le pretende dar y que sea acorde a los usuarios a los que va destinado se puede contar con la participación de usuarios en el proceso de desarrollo. Para ello el modo de trabajo puede tener que modificarse para acomodar a unos miembros del equipo con escasa o nula formación en Ingeniería del Software. Nuevamente, el uso de la técnica de prototipado parece la más adecuada para hacer participar a representantes de usuarios en el diseño del sistema.

Finalmente, es preciso destacar que si se opta por un ciclo de vida puro en cascada resulta prácticamente imposible aplicar satisfactoriamente técnicas de usabilidad en el desarrollo del sistema.

¿Por qué Aplicar Ingeniería de Usabilidad?

La principal razón por la cual aplicar la Ingeniería de Usabilidad cuando se desarrolla un sistema software, es la obtención de un sistema que hace al usuario más productivo, aumentando su eficiencia y satisfacción al utilizarlo.

La usabilidad es un tema crítico para la aceptación de un sistema: Si el sistema no es percibido como una herramienta que ayuda al usuario a realizar sus tareas, se dificulta la aceptación del sistema. Puede ocurrir que el sistema no llegue a usarse en absoluto, o que se use con escasa eficiencia. Si las tareas del usuario no son respaldadas convenientemente por el sistema, entonces no se está respondiendo adecuadamente a las necesidades del usuario, y el equipo de desarrollo se está alejando del objetivo principal de la construcción de un sistema software.

De cara a la organización de desarrollo de software también es importante invertir en la usabilidad de los sistemas que construye. Puede ocurrir que se detecten errores graves de usabilidad en un momento cercano al final estimado del proyecto, y entonces el tiempo de desarrollo puede crecer en demasía para corregir tales problemas. Por otra parte, hay que tener en cuenta los elevados costes del servicio de atención al usuario que pueden derivarse de un sistema con deficiencias graves de usabilidad. En un mercado altamente competitivo también puede ser una seria amenaza el lanzamiento por parte de un competidor de un sistema con un nivel alto de usabilidad, y frente a tal posibilidad es muy importante invertir en usabilidad para poder mantener o ampliar la cuota de mercado.

Técnicas de Ingeniería de Usabilidad

En esta sección se van a exponer las principales técnicas que ayudan a conseguir un mejor nivel de usabilidad en el software desarrollado.

Como se ha indicado anteriormente, es preciso que se siga un ciclo de vida iterativo. En una fase previa se establecen unas especificaciones de usabilidad que van a servir de patrón con el que comparar el nivel de usabilidad del sistema. A continuación comienza un ciclo diseño-evaluación-rediseño que finaliza cuando se alcanzan los niveles detallados en las especificaciones de usabilidad.

Las técnicas que se exponen a continuación van a agruparse según su uso en dicho ciclo:

- Especificaciones: Análisis de usuarios, análisis de tareas y especificaciones de usabilidad.
- Diseño: Diseño de la interacción, prototipado y participación de usuarios.
- Evaluación: Test de usabilidad y evaluación heurística.

Para cada técnica o concepto se va a explicar su motivación, el procedimiento que siguen y las técnicas relacionadas, si corresponde.

Especificaciones

Al principio del proyecto se elaboran unas especificaciones de usabilidad intentando que realmente reflejen el nivel de usabilidad del sistema en los aspectos específicos que más interesen. Estas especificaciones dirigirán el proceso iterativo de desarrollo, pero para crearlas será necesario identificar previamente a los usuarios y las tareas que van a desarrollar con el sistema. Esta parte está muy relacionada con la Ingeniería de Requisitos.

Análisis de Usuarios

Si se desea construir un sistema software usable, se debe primero conocer a fondo a qué usuarios específicos está destinado, cuáles son sus características principales [Hix93].

Motivación: Para conocer a los usuarios, las tareas que desarrollan y cómo las llevan a cabo. Es importante conocer cómo piensa el usuario para desarrollar un sistema que trabaja según ese esquema (y no según el esquema mental del equipo de desarrollo).

Procedimiento: Depende del sistema específico a desarrollar. Algunos procedimientos que pueden llevarse a cabo son los siguientes:

- Análisis de mercado: Adecuado para software comercial.
- Visitas de campo: Cuando se desarrolla software para una empresa u organización es muy útil observar al usuario en su entorno de trabajo habitual, más si cabe en el caso de que el usuario esté utilizando un sistema que será reemplazado por el que se va a construir.
- Cuestionarios: Sin ser tan útil como hablar a los usuarios en su entorno habitual, la información acerca de los usuarios puede ser recogida mediante cuestionarios.

Técnicas relacionadas: El análisis de usuarios puede proporcionar una categorización de usuarios, la cual puede ser útil a la hora de obtener una muestra de usuarios con los que realizar test de usabilidad.

Análisis de Tareas

El término *Análisis de Tareas* se usa para describir un conjunto de técnicas que se preocupan de cómo hace la gente para realizar una determinada tarea [Preece94]. El concepto de tarea es similar al de función, pero no igual. Una tarea es una actividad con sentido para el usuario, algo que el usuario considera necesario o deseable que se realice.

Motivación: Descomponer la interacción con el sistema en unidades con sentido para el usuario. Estas unidades serán el punto de partida a la hora de desarrollar el sistema.

Procedimiento: Si en la actualidad la población de usuarios ya desarrolla una serie de tareas, se realiza un análisis de tareas actuales durante el análisis de usuarios, para identificar dichas tareas y el modo en que el usuario las percibe. Tras este primer análisis opcional, se identifican las tareas que el sistema a desarrollar va a ofrecer, en base a los objetivos que el usuario quiere satisfacer. Las tareas son descompuestas entonces en subtareas y éstas en acciones que el usuario realizará en su interacción con el sistema.

Técnicas relacionadas: El análisis de usuarios se toma como entrada al análisis de tareas, y ambos se realizan conjuntamente en determinadas ocasiones. El conjunto de tareas que se obtienen sirve como base para la elaboración de las especificaciones de usabilidad, y se instancian a ejemplos reales para que las lleven a cabo los participantes en los test de usabilidad.

Especificaciones de Usabilidad

Se establecen especificaciones de usabilidad como objetivos cuantitativos de usabilidad [Whiteside88], los cuales se definen antes de comenzar con el diseño del sistema. Se basan en los cinco atributos de usabilidad básicos descritos anteriormente, o en subatributos de los mismos.

Motivación: Si se quiere poder medir la usabilidad del sistema que se está construyendo, es preciso tener un conjunto de especificaciones de usabilidad que puedan ser verificadas.

Procedimiento: Se define un conjunto de especificaciones de usabilidad para cada atributo de usabilidad que se quiera medir (esto es, que sea importante para el sistema a construir). Deben definirse de modo que puedan medirse, bien mediante test de usabilidad, bien mediante cuestionarios.

Técnicas relacionadas: La mayor parte de las especificaciones de usabilidad están asociadas a una tarea específica de las obtenidas en el análisis de tareas. Los test de usabilidad nos dan los valores de las especificaciones para la versión actual del sistema que se está desarrollando.

Tabla 1. Ejemplo del formato de una tabla de especificaciones de usabilidad (basada en [Hix93]).

Atributo de Usabilidad	Medio de Medición	Valor a Medir	Nivel Actual	Peor Nivel Aceptable	Nivel Planificado como Objetivo	Mejor Nivel Posible	Resultados Observados
Rendimiento en uso normal	Tarea “Atender petición”	Tiempo empleado en llevar a cabo correctamente la tarea (minutos y segundos)	2’ 53’’	2’ 53’’	1’ 30’’	50’’	
Primera impresión	Cuestionario	Puntuación media (rango -2 a 2)	-	0	1	2	

Diseño

Una vez identificadas las tareas a las que el sistema va a dar soporte, se puede empezar a diseñar la interacción del sistema, como una primera aproximación que será evaluada y, eventualmente, mejorada en posteriores iteraciones.

En primer lugar se va a describir el diseño de la interacción del sistema y las técnicas de prototipado relacionadas con la usabilidad. A continuación se van a describir dos principios de diseño que implican en distinto grado al usuario en el diseño del sistema: el diseño centrado en el usuario y el diseño participativo.

Diseño de la Interacción

El diseño de la interacción se puede dividir en dos etapas: Diseño del concepto del sistema y diseño de la parte visual de la interacción.

El diseño del concepto del sistema es la actividad más importante del desarrollo de software, pues definirá de qué modo va a funcionar el sistema. Es importante crear un concepto del sistema que pueda ser comprendido y asimilado fácilmente por el usuario. Para ello se pueden usar metáforas (como la metáfora del escritorio usada por algunos sistemas operativos) basadas en conceptos familiares para el usuario, o bien imitar sistemas ya conocidos por el usuario.

El diseño es una actividad creativa y no puede mecanizarse. De todos modos, aunque no hay recetas de cómo crear un buen concepto del sistema, sí hay principios generales que nos pueden guiar en dicha tarea, como intentar lograr una consistencia en la interacción, intentar minimizar la posibilidad de error por parte del usuario, no sobrecargar la memoria del usuario, ofrecer realimentación al usuario sobre sus acciones, etc. En el trabajo de Constantine y Lockwood [Constantine99] se puede encontrar una gran cantidad de consejos de diseño.

El concepto del sistema se materializa posteriormente al realizar el diseño de la parte visual de la interacción (interfaz gráfica de usuario). Hay una serie de normas pertenecientes al campo del diseño gráfico sobre cómo escoger los colores, tipos de letra, la disposición de los elementos en una ventana, etc. Esta parte suele realizarla un diseñador gráfico profesional.

Normalmente el diseño desemboca en la creación de un prototipo para ser evaluado con usuarios.

Prototipado

El prototipado no es una técnica exclusiva de la Ingeniería de Usabilidad, pero es muy valiosa en las primeras fases del desarrollo para representar el diseño de la interacción y evaluar su usabilidad.

Motivación: No es posible conocer la opinión de los usuarios mostrándoles especificaciones técnicas a un nivel abstracto. Los usuarios entenderán mucho mejor prototipos concretos del sistema.

Procedimiento: Algunas técnicas de prototipado ayudan a representar la interacción con un esfuerzo mínimo de implementación:

- *Borradores en papel:* Al principio del proceso de diseño se pueden crear prototipos sobre papel para mostrarlos al usuario. Normalmente son representaciones de las ventanas de la aplicación. El diseñador actúa como sistema, presentando al usuario el siguiente elemento cuando ocurre una transición entre ventanas [Nielsen93].
- *Técnica del Mago de Oz* [Preece94]: Un experto humano actúa como sistema, dando las respuestas a las peticiones del usuario. El usuario interactúa normalmente con el terminal, pero en vez de haber un

software detrás, un desarrollador está frente a otro terminal conectado al primero a través de la red, y va tecleando la supuesta respuesta del sistema. El usuario normalmente no sabe acerca del “truco”, para conseguir la sensación de estar tratando con un sistema real.

- *Escenarios, storyboards y viñetas*: Un escenario describe una historia de ficción de un usuario interactuando con el sistema en una situación concreta. Las viñetas son representaciones que capturan la interacción que ocurre en un escenario. *Storyboards* son secuencias de viñetas que se centran en las principales acciones en una situación dada [Preece94]. Estas técnicas posibilitan que el equipo de diseño piense acerca de lo apropiado del diseño actual a las necesidades del usuario, favoreciendo un proceso de diseño más centrado en el usuario.

Participación de los Usuarios

Para que el sistema que está diseñando sea realmente como quieren y/o necesitan los usuarios se puede involucrar a éstos para que participen en el proceso de diseño. Esta filosofía de diseño se conoce como Diseño Centrado en el Usuario. El Diseño Participativo va un paso más allá y pone a representantes de usuarios como responsables de decisiones de diseño.

Diseño Centrado en el Usuario

Es la aplicación al diseño de la filosofía en la que se basa la Ingeniería de Usabilidad. Se trata de centrar el proceso de diseño en el usuario, implicando a los usuarios en el diseño para confirmar que se está desarrollando un sistema que en realidad satisface sus necesidades. La interacción debe ser diseñada desde el punto de vista del usuario [Hix93].

Se desea obtener sistemas que sean fáciles de aprender y de utilizar, y efectivos para las tareas a las que dan soporte. Para ello el punto clave es el usuario, y no se puede conocer su punto de vista sin involucrarle en el proceso de desarrollo. Tal participación se lleva a cabo principalmente mediante la creación de prototipos a lo largo de todo el proceso de desarrollo, para que un grupo representativo de usuarios los evalúe.

Diseño Participativo

También se conoce como *Diseño escandinavo*, debido a que surgió en los países escandinavos y es su principal ámbito geográfico de aplicación. Es una aproximación al diseño que pertenece no sólo al diseño de sistemas software, sino también al diseño industrial. Este enfoque acepta la tesis de que el usuario es el máximo experto en diseñar un sistema que satisfaga sus necesidades. Los usuarios son los diseñadores del sistema, actuando los ingenieros software como consejeros técnicos que indican qué se puede y qué no se puede hacer [Schuler93]. El papel de los usuarios no va a ser el de un elemento pasivo al que se consulta sobre temas específicos, sino como núcleo central del equipo de desarrollo.

Esta aproximación es factible únicamente cuando el clima político y la cultura empresarial de la organización es el apropiado [Preece94], y es preciso también que los representantes de los usuarios estén disponibles por un periodo de tiempo significativo.

Cuando hay usuarios participando en el equipo de desarrollo es preciso utilizar medios y lenguaje no técnicos. Por ejemplo, el diseño de la interfaz mediante notas *Post-It* es un medio sencillo y flexible para construir borradores en papel.

Evaluación

La evaluación de usabilidad permite conocer el nivel de usabilidad que alcanza el prototipo actual del sistema, e identificar los fallos de usabilidad existentes.

La evaluación se realiza usualmente mediante test de usabilidad, complementados con evaluaciones heurísticas.

Test de Usabilidad

Los test de usabilidad son la práctica de usabilidad más extendida. Consisten en presentar al usuario una serie de tareas a realizar, y pedirle que las realice con el prototipo del sistema. Las acciones y comentarios de usuario se recopilan para un análisis posterior.

Para conseguir unos test de usabilidad con resultados fiables, las condiciones del test y del lugar donde éste se realiza deben ser lo más parecidas posibles al entorno de uso previsto para el sistema.

Motivación: Se basa en que no es posible conocer el nivel de usabilidad de un sistema si no se prueba con usuarios reales.

Procedimiento: En primer lugar, es preciso decidir con qué distintos grupos de usuarios se va a probar el sistema, y cuántos participantes de cada grupo se van a obtener para realizar los test. A continuación, se deben

diseñar las tareas de test cuya realización se va a pedir a los participantes; normalmente se sacan del resultado del análisis de tareas, intentado enmarcarlas en un contexto de uso real. Hay que decidir también otros detalles, como la posibilidad de pedir ayuda al evaluador, qué tipo de información se dará a los participantes acerca del sistema antes de comenzar con el test en sí, o si se dará la posibilidad al participante de acceder libremente al sistema para obtener opiniones acerca de su impresión global. Otra variante consiste en realizar cada test con dos usuarios para observar los comentarios que intercambian. Es habitual pedir al participante que “piense en voz alta” mientras intenta llevar a cabo las tareas; este procedimiento permite identificar partes del sistema con un nivel pobre de usabilidad.

Cuando el test está preparado y los participantes han sido reunidos, los test se llevan a cabo, opcionalmente grabados en audio o vídeo. Otra posibilidad es registrar las acciones de los usuarios en un fichero del sistema para un análisis posterior.

Una vez se han realizado todos los test, los datos recogidos son analizados y los resultados se aplican en el siguiente ciclo de diseño.

Evaluación Heurística

Un experto en usabilidad (o en HCI) puede realizar una evaluación heurística del sistema para hacer algunas iteraciones de desarrollo más cortas, y así ser capaces de realizar más iteraciones en el proceso de desarrollo. El experto realizará una crítica basado en su experiencia de diseño de la interacción, o en guías de diseño de usabilidad ampliamente aceptadas, como las descritas por Shneiderman [Shneiderman98] o Nielsen [Nielsen93].

Los expertos proporcionan una información distinta a la obtenida de usuarios finales mediante test de usabilidad. Las sugerencias de modificaciones por parte de un experto suelen tener más valor que las realizadas por usuarios, por ser más viables y más precisas acerca de los problemas subyacentes de usabilidad (falta de consistencia, navegabilidad pobre, etc.) Por otra parte, para identificar problemas de usabilidad específicos es preciso realizar test de usabilidad con usuarios reales. La evaluación heurística no debe usarse en vez de los test de usabilidad, sino como complemento a los mismos.

Motivación: Los expertos identifican ciertos problemas de usabilidad que requerirían un gran número de test de usabilidad para ser correctamente identificados y solucionados.

Procedimiento: Se explica al experto el modo de funcionamiento del sistema, y se le entrena en las funciones principales. Se le informa acerca del dominio de aplicación y el experto revisa el sistema según la conformidad del mismo a guías de diseño. Una vez finalizada la revisión el experto elabora un informe con los problemas identificados y sugerencias de diseños alternativos (opcionalmente), el cual es entregado al equipo de desarrollo.

Cómo Empezar a Aplicar Técnicas de Usabilidad

En el caso de que un ingeniero software esté interesado en aplicar los conceptos y técnicas descritos anteriormente en el software que desarrolla, normalmente no intentará aplicar todo de una vez. A continuación se muestran una serie de acciones que se pueden tomar para empezar a implantar la preocupación por la usabilidad en una organización de desarrollo de software.

La primera tarea a realizar consiste en obtener apoyo de la dirección a los principios de Ingeniería de Usabilidad, convenciéndoles acerca de los beneficios que se obtienen al aplicarlos [Nielsen93][Hix93]. El concepto que, probablemente, será más novedoso para la dirección será la necesidad de llevar a cabo evaluación de usabilidad a lo largo de todo el proceso de desarrollo, incluyendo las primeras fases. En la obra de Bias y Mahew [Bias94] se detallan una serie de argumentos coste/beneficio a favor de la implantación de técnicas de usabilidad en una organización de desarrollo de software.

Se puede tomar un sistema que esté actualmente en desarrollo, o que haya sido desarrollado recientemente, y realizar una serie de test de usabilidad grabados en vídeo con unos pocos usuarios que no hayan utilizado previamente el sistema. Mostrando una selección de las escenas grabadas a la dirección y/o al equipo de desarrollo puede producir un cambio de actitud hacia los test de usabilidad, puesto que los resultados probablemente mostrarán que la usabilidad del sistema no es tan buena como se pensaba.

En vez de realizar test de usabilidad a gran escala desde el principio, es mejor empezar por establecer un pequeño conjunto de especificaciones de usabilidad con un sencillo análisis de tareas que detalle las principales de entre ellas, y preparar test de usabilidad simples para ser llevados a cabo con un número reducido de usuarios. También se puede actuar como experto en usabilidad llevando a cabo evaluaciones heurísticas del sistema. Empezando con objetivos modestos se podrá conseguir más fácilmente la introducción paulatina de los principios de usabilidad.

Nielsen ha introducido el concepto de “Usabilidad de Rebajas” (*Discount Usability Engineering* en inglés) [Nielsen93], que propone realizar sólo algunas de las más sencillas técnicas de Ingeniería de Usabilidad cuando los recursos son escasos. El principio en el que se basa esta idea es que aplicando estas técnicas “baratas”, se

puede conseguir una importante mejora en el nivel de usabilidad del sistema software desarrollado, a pesar de que se obtendría un resultado mejor con la aplicación de todas las técnicas a gran escala. Las técnicas que propone aplicar son: Observación de usuarios y tareas, prototipos que implementan sólo la parte del sistema necesaria para llevar a cabo una tarea bajo condiciones específicas, test de usabilidad con “pensar en voz alta” simplificado, y evaluación heurística.

Integración entre Usabilidad e Ingeniería del Software

Como conclusión se van a describir las dificultades que existen a la hora de integrar la Ingeniería de Usabilidad con la Ingeniería del Software. Se trata de un tema que no está todavía resuelto.

Los ingenieros software han visto tradicionalmente la usabilidad como análoga a otros atributos que se prueban en test de aseguramiento de calidad, normalmente al final del proceso de desarrollo. Esta situación ha llevado a la aplicación de las técnicas de usabilidad muy tarde en el ciclo de desarrollo, cuando los principales problemas de usabilidad son demasiado costosos de arreglar. Puesto que los principales problemas de usabilidad son fácilmente detectables en las primeras fases del desarrollo, hay una tendencia a integrar la Ingeniería de Usabilidad en el proceso de desarrollo de una forma más compacta [Trenner98].

Tal integración no resulta fácil, pues hasta ahora ambos procesos de desarrollo (el de Ingeniería de Usabilidad y el de Ingeniería del Software) discurrían casi independientemente, llevados a cabo por equipos distintos. Los principales problemas a solucionar son los siguientes:

- A pesar de que los ingenieros software se pueden mostrar receptivos a los principios de usabilidad, su formación y experiencia hace que entre sus principales objetivos estén conceptos como la elegancia de los algoritmos diseñados, el rendimiento a bajo nivel o la facilidad de mantenimiento, los cuales en ocasiones influyen negativamente en la usabilidad del sistema.
- Los conceptos que se manejan en la Ingeniería de Usabilidad son ajenos a los manejados en la Ingeniería del Software, y esto produce un problema de comunicación entre el equipo de expertos en usabilidad y el resto del equipo de desarrollo.
- Es necesario realizar una integración de la Ingeniería de Usabilidad en la parte de planificación y gestión de proyectos software.

Bibliografía

- [Bias94] R. G. Bias, D. J. Mayhew. *Cost-Justifying Usability*. Academic Press, 1994.
- [Constantine99] L. L. Constantine, L. A. D. Lockwood. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley, New York, NY, 1999.
- [Hewett92] T. Hewett. *ACM SIGCHI Curriculum for Human-Computer Interaction*. ACM Press, 1992.
- [Hix93] D. Hix, H.R. Hartson. *Developing User Interfaces: Ensuring Usability Through Product and Process*. John Wiley and Sons, 1993.
- [ISO98a] *ISO 14598-1. Information Technology – Evaluation of Software Products – General Guide*. ISO, 1998.
- [ISO98b] *ISO 9241-11. Ergonomic requirements for office work with visual display terminals*. ISO, 1998.
- [ISO00] *ISO Final Draft International Standard 9126-1. Software Engineering – Product Quality – Part 1: Quality Model*. ISO, 2000.
- [Nielsen93] J. Nielsen. *Usability Engineering*. AP Professional, 1993.
- [Preece94] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey. *Human-Computer Interaction*. Addison Wesley, 1994.
- [Schuler93] D. Schuler, A. Namioka. *Participatory Design: Principles and Practices*. Lawrence Erlbaum Associates, 1993.
- [Shneiderman98] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 1998.
- [Trenner98] L. Trenner, J. Bawa. *The Politics of Usability*. Springer-Verlag, 1998.
- [Whiteside88] J. Whiteside, J. Bennett, K. Holtzblatt. “Usability Engineering: Our Experience and Evolution”. *Handbook of Human-Computer Interaction*. Elsevier North-Holland, 1988.