



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

03063
/

**EVALUACIÓN DE INTERFACES DE USUARIO
EN EL PROCESO UNIFICADO**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS
(Computación)**

P R E S E N T A:

RODRIGO FERNANDO ALONSO PINZÓN

DIRECTOR DE TESIS: DR. FERNANDO GAMBOA RODRÍGUEZ

MÉXICO, D.F.

2002.

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: _____

Rodrigo F. Abiza Pinzón

FECHA: _____

27/10/2007

FIRMA: _____



ESTA TESIS NO SALE
DE LA BIBLIOTECA

RECIBO
ALBINO TO ALBINO

Dedicatorias y agradecimientos

A mi familia, Jorge, Maricela, Leonora y Elías. La mejor familia del planeta.

A mi abuelo, Fernando Alonso. Un excelente maestro

A mis amigos Juan, Eduardo, Martín y Eduardo. Los mejores amigos que hay

A mi tío Fernando y mi primo Fernando. Siempre importantes

Agradezco al Dr. Fernando Gamboa, a la Dra. Hanna Oktaba y a la Mstra Guadalupe Ibarguengoitia por sus excelentes clases y por la formación que me dieron.

Finalmente a la Universidad Nacional Autónoma de México por proporcionarme siete años de excelente formación y ambiente de desarrollo personal.

Índice General

| | | |
|-----------|--|------------|
| I | Introducción | vii |
| 1 | Motivaciones | 1 |
| 1.1 | Algunas consideraciones | 1 |
| 1.2 | El consumidor y la IDS | 4 |
| 1.3 | El usuario actual | 5 |
| 1.3.1 | Un ser exigente | 5 |
| 1.3.2 | El software desde el punto de vista del usuario | 7 |
| 1.4 | La motivación | 8 |
| 1.5 | Los objetivos | 9 |
| 2 | Ingeniería de software actual | 11 |
| 2.1 | Breve historia de la Ingeniería de software | 11 |
| 2.1.1 | La era pionera | 12 |
| 2.1.2 | Dominando los procesos | 13 |
| 2.1.3 | Dominando la complejidad | 15 |
| 2.1.4 | El presente intento de formalización | 17 |
| 2.2 | El estado actual | 17 |
| 2.3 | El objetivo | 19 |
| 3 | Interacción humano-computadora | 21 |
| 3.1 | Interacción humano-computadora | 21 |
| 3.2 | Evolución e importancia de IHC | 23 |
| 3.3 | IHC en software | 25 |
| II | Pruebas de usuario e ingeniería de software | 29 |
| 4 | Ingeniería de software y usabilidad | 31 |

| | | |
|------------|---|-----------|
| 4.1 | Sobre los procesos de desarrollo | 31 |
| 4.2 | El usuario y la interfaz | 32 |
| 4.3 | Usuario e interfaz en los procesos de desarrollo | 33 |
| 4.3.1 | El Proceso Unificado | 33 |
| 4.3.2 | Extreme Programming | 38 |
| 4.3.3 | Team Software Process | 39 |
| 4.3.4 | Otras referencias | 41 |
| 4.4 | La necesidad de evaluar interfaces dentro de los procesos | 43 |
| 5 | Pruebas de usuario | 45 |
| 5.1 | Breve introducción al Desarrollo Centrado en el Usuario | 45 |
| 5.2 | ¿Qué es una prueba de usuario? | 46 |
| 5.3 | La importancia de las pruebas de usabilidad | 48 |
| 5.3.1 | La importancia para el cliente | 49 |
| 5.3.2 | La importancia para el desarrollador | 50 |
| 5.3.3 | La importancia para el usuario | 51 |
| 5.4 | ¿Cómo llevar a cabo una prueba de usuario? | 52 |
| 5.4.1 | Las pruebas de usabilidad, los distintos tipos | 52 |
| 5.4.2 | El prototipo | 53 |
| 5.4.3 | Pruebas exploratorias | 55 |
| 5.4.4 | Pruebas de diseño | 56 |
| 5.4.5 | Pruebas de desempeño | 57 |
| 5.4.6 | Preparación para la prueba | 58 |
| 5.4.7 | Las seis etapas de la prueba | 60 |
| III | Pruebas de usuario y Proceso Unificado | 67 |
| 6 | El Proceso Unificado y la interfaz | 69 |
| 6.1 | La interfaz de usuario en el Proceso Unificado | 69 |
| 6.2 | Diseño de la interfaz de usuario | 70 |
| 6.2.1 | De las reglas del negocio a la interfaz | 70 |
| 6.2.2 | El diseño físico | 73 |
| 6.2.3 | Las pruebas | 74 |
| 7 | Pruebas y Proceso Unificado | 77 |
| 7.1 | Las pruebas a lo largo de todo el proceso | 77 |
| 7.2 | Artefactos, Trabajadores y Actividades | 78 |

| | | |
|----------|---|------------|
| 7.3 | Artefactos | 79 |
| 7.3.1 | Artefacto: Hipótesis de usabilidad | 79 |
| 7.3.2 | Artefacto: Plan de pruebas de usabilidad | 80 |
| 7.3.3 | Artefacto: Cuestionario de perfil del usuario | 80 |
| 7.3.4 | Artefacto: Protocolo de bienvenida | 81 |
| 7.3.5 | Artefacto: Tareas de usuario | 81 |
| 7.3.6 | Artefacto: Instrumento | 82 |
| 7.3.7 | Artefacto: Cuestionario de usabilidad | 83 |
| 7.3.8 | Artefacto: Registro de prueba de usabilidad | 83 |
| 7.3.9 | Artefacto: Observaciones de prueba de usuabilidad | 84 |
| 7.3.10 | Artefacto: Resultados de prueba de usabilidad | 84 |
| 7.4 | Trabajadores | 87 |
| 7.4.1 | Trabajador: Ingeniero de pruebas | 87 |
| 7.4.2 | Trabajador: Monitor de pruebas de usabilidad | 87 |
| 7.4.3 | Trabajador: Diseñador de interfaces de usuario | 88 |
| 7.4.4 | Trabajador: Analista de pruebas de usabilidad | 88 |
| 7.5 | Actividades | 88 |
| 7.5.1 | Actividad: Planear prueba de usabilidad | 89 |
| 7.5.2 | Actividad: Diseñar prueba de usabilidad | 90 |
| 7.5.3 | Actividad: Realizar prueba de usabilidad | 90 |
| 7.5.4 | Actividad: Observar prueba de usabilidad | 92 |
| 7.5.5 | Actividad: Analizar prueba de usabilidad | 94 |
| 7.6 | Diferencias entre flujos de trabajo | 96 |
| 7.6.1 | Inicio y Elaboración | 96 |
| 7.6.2 | Construcción | 97 |
| 7.6.3 | Transición | 98 |
| 7.7 | Invariancia del Proceso Unificado | 99 |
| 7.7.1 | La estrucutra actual | 99 |
| 7.7.2 | La nueva estructura | 103 |
| 8 | Caso de prueba: Proyecto SIC | 105 |
| 8.1 | El proyecto | 106 |
| 8.2 | Las pruebas | 106 |
| 8.2.1 | Los participantes | 106 |
| 8.2.2 | El lugar | 107 |
| 8.2.3 | Los trabajadores | 107 |
| 8.3 | Los artefactos | 109 |
| 8.3.1 | Hipótesis de usabilidad | 109 |

| | | |
|----------|------------------------------------|------------|
| 8.3.2 | Plan de pruebas de usabilidad | 113 |
| 8.3.3 | Cuestionario de perfil del usuario | 115 |
| 8.3.4 | Protocolo de bienvenida | 119 |
| 8.3.5 | Tareas de usuario | 120 |
| 8.3.6 | Instrumento | 123 |
| 8.3.7 | Cuestionario de usabilidad | 124 |
| 8.3.8 | Observaciones de usabilidad | 128 |
| 8.3.9 | Resultados de usabilidad | 130 |
| 8.4 | Resultados | 146 |
| 9 | Conclusiones | 149 |
| 9.1 | Las pruebas de usabilidad | 150 |
| 9.1.1 | Los resultados | 151 |
| 9.1.2 | Evaluación de las pruebas | 151 |
| 9.2 | El panorama | 152 |
| 9.3 | Un mejor proceso | 153 |
| 9.4 | Conclusiones finales | 154 |
| A | Artefactos de las pruebas | 155 |
| A.1 | Plan de pruebas de usabilidad | 156 |
| A.2 | Cuestionario de perfil del usuario | 157 |
| A.3 | Protocolo de bienvenida | 158 |
| A.4 | Tareas de usuario | 159 |
| A.5 | Especificación de los instrumentos | 160 |
| A.6 | Cuestionario de usabilidad | 161 |
| A.7 | Reporte general de usabilidad | 163 |
| A.8 | Reporte de desempeño | 165 |
| A.9 | Recomendaciones de usabilidad | 165 |
| B | Acrónimos | 167 |

Parte I
Introducción

Capítulo 1

Motivaciones

1.1 Algunas consideraciones

De entre todas las industrias que existen destacaremos una, la destacaremos no sólo por lo singular del producto que genera, sino también por lo singular de su estructura interna. Adicionalmente a esta singularidad, esta industria se ha vuelto una de las industrias más importantes a nivel mundial a pesar de ser extremadamente joven, unas tres décadas. La industria a la que hacemos referencia es la *Industria del software* (IDS), la creadora de uno de los tipos de producto más complejos que existen y una de las responsables de que la humanidad haya entrado en la era de la información.

Ahora, cuando se piensa en la palabra “industria”, surgen en la mente palabras como “organización” y “automatización”, se piensa en algo que satisface una necesidad del ser humano y que generalmente proporciona algo con lo que el consumidor está contento. Bien, como veremos, la IDS frecuentemente proporciona algo al consumidor con lo que éste no está contento. Antes de plantear el alcance y motivaciones del presente trabajo, comentaremos algunas ideas más acerca de la IDS que servirán como marco de referencia de futuras discusiones.

Los productos ingenieriles pueden ser simples o extremadamente complejos, las industrias que se encargan de crear estos productos lo hacen echando mano de técnicas de automatización, una buena organización y formalismo dependiendo del producto del cual se trate. En general, entre más complejo

sea un producto, más formalismo será usado en su diseño¹. Automatización, organización, formalismo. Estas tres palabras son capitales en una industria y serán mencionadas varias veces en este texto; son lo que hace a la industria misma y lo que la ha llevado hasta su nivel actual de producción.

Desde sus inicios, la industria tuvo que encontrar maneras de organizarse de tal manera que pudiera cubrir una demanda creciente de productos, mismos que constantemente tenían que tener mejor calidad y ser más baratos. La industria ha cambiado por necesidad.

Para entender el contexto en el que entra la IDS tenemos que pensar un poco en cómo funciona la industria común. Cada individuo en una industria tiene un trabajo que hacer, mismo que está trazado por los planos del producto. Los roles de los trabajadores son inambiguos, se tiene una organización que se respeta en el tiempo. Así mismo, los productos pasan primero por una etapa de diseño, en la cual se modela el futuro producto, una etapa de construcción, una etapa de prueba (en la cual, dependiendo de la importancia que tiene el usuario, éste puede estar muy involucrado), y una etapa de producción. Actualmente cuando un producto pasa la etapa de pruebas, se espera que sea de la totalidad satisfacción del usuario.

Los productos de software son creados por la IDS, una industria especial. Pensemos en las características que hemos mencionado en los párrafos anteriores: la IDS no tiene una línea de producción, pero ¿Qué hay acerca de la automatización, la organización y el formalismo? Una empresa de software tiene una estructura y una organización interna bien definida, pero razonemos esto.

Algo es cierto en general solamente si es cierto para cada caso. La IDS sigue procesos para crear software, sin embargo, al contrario de otras industrias en las que se sigue un proceso similar (como la industria automotriz), una empresa de software en particular puede seguir muchas diferentes propuestas de procesos. Entonces puede afirmarse que, si bien un producto de software es creado generalmente siguiendo un proceso bien definido, la IDS en general no tiene una organización única.

¹Cuando se expresa algo de manera *formal*, se dice que se expresa mediante formas y métodos previamente establecidos o acordados.

¿Qué hay de la automatización? Automatización significa hacer que máquinas realicen el trabajo previamente realizado por humanos. La revolución industrial sucedió gracias a la automatización y podríamos pensar que cualquier industria que se precie de serlo, actualmente tiene que tener alguna forma de automatizar procesos. Bien, sucede que la IDS, a pesar de sus tres décadas de existencia, todavía está en vías de automatizar procesos *clave*s dentro de lo que es “El proceso de construcción del software”.

¿Y qué hay del formalismo? Bien, dejando a un lado las matemáticas y tomando en cuenta la definición dada anteriormente (pie de página número 1), se puede decir que formalizar algo quiere decir ponerlo en forma escrita y clara mediante un “lenguaje” previamente establecido. Aquí “escrita” se refiere a cualquier forma de plasmar modelos, diseños o instrucciones de forma que cualquier persona dentro de una misma disciplina pueda entender. Aquí también entra la claridad, para cualquier músico una partitura debe ser igual de clara así como para cualquier arquitecto cualquier plano debe poder ser entendido. Como se muestra a continuación, el software es diferente, es una rama de la ingeniería que carece de formalismo [Hamlet, Maybee, 2001].

Durante un proceso completo de desarrollo de software puede llegar a acumularse una cantidad insana de documentación. Toda esta documentación explica hasta los más mínimos detalles, tanto del proceso de construcción como del producto mismo. ¿Con qué lenguaje y en qué forma está escrita esta documentación? ¿Acaso cualquier ingeniero de software es capaz de entender los distintos diagramas que explican el comportamiento dinámico de cierta parte de un sistema? Como veremos más adelante, la formalización durante el proceso de desarrollo de software existe como propuesta desde hace algunos años. Aunque ya habían diferentes propuestas, debido a que ninguna de ellas fue adoptada como estándar por la IDS, una propuesta unificada debe considerarse como algo nuevo.

Ahora que hemos analizado brevemente ciertos aspectos importantes de la IDS actual (automatización, organización, formalismo), pasaremos a otro aspecto que es igual de importante y que es consecuencia de algunas de las cuestiones antes mencionadas.

1.2 El consumidor y la IDS

Sin duda hay industrias que generan cierta antipatía por parte de los consumidores, pero si hay una que es el blanco de constantes críticas es la IDS. Frecuentemente el consumidor no obtiene lo que espera, sobre todo cuando acaba de pagar cientos y hasta miles de dólares por un producto de software. Por supuesto que la IDS lo sabe y sabe también que tiene que cambiar. Ahora, para darle al consumidor algo de su completo agrado, que amerite el precio que pagó y que sea fácil de usar y confiable, se necesita cambiar a la IDS si no en todos sus aspectos, ciertamente sí en algunos.

Muchas industrias, desarrollan sus productos centrándose en las necesidades del usuario², i.e, llevan a cabo un *desarrollo centrado en el usuario* (DCU). Después de todo, el agrado y comodidad del usuario dicta el éxito de un producto. Que una empresa o industria ponga mucho énfasis en el agrado del usuario denota aspectos importantes de la misma. Esto es, por lo menos se tienen resueltos muchos de los aspectos que tienen que ver con la organización de su industria, la construcción y el diseño de sus productos. Es entonces lógico que puedan darse a la tarea de satisfacer necesidades estéticas y de usabilidad: Esto está dado por la madurez de la industria misma.

¿Cómo puede dedicar esa misma cantidad de recursos para agradar al usuario la IDS cuando muchos de los aspectos de la construcción misma del producto no han sido totalmente resueltos? ¿Será esta una de las razones por las cuales agradar al usuario no figura en muchos de los procesos y estándares de desarrollo de software?

En efecto, cuando se tienen las necesidades básicas satisfechas se puede empezar a atender los demás aspectos. Es por eso que la IDS ha estado en busca de una mejor manera de crear software. Un proceso que ayude a satisfacer estas necesidades primarias para empezar a tener un flujo constante de productos de calidad y entre cuyas características figure el agrado del usuario

Pero esto trae consigo otro tipo de problemas: con tantos tipos diferentes de procesos de desarrollo que han surgido, certificaciones y estándares que

²La industria del calzado deportivo y la de los muebles para oficina son un par de ejemplos.

hay que seguir ¿A cuáles se adhiere una empresa de software en particular? Y sobre todo ¿El seguir un determinado proceso de desarrollo o apegarse a un estándar, garantiza que el problema de organización dentro de una empresa quedará resuelto? ¿Garantiza eso que se producirá una pieza de software de calidad? ¿El usuario tendrá lo que espera? Aún cuando las respuestas a estas preguntas no son sencillas, lo que sí podemos afirmar es que seguir un buen proceso de desarrollo es importante, pero ¿Cuál seguir? ¿Qué experiencia respalda a cada uno?

Muchas preguntas, pero toda empresa de software que piense seriamente en entrar al negocio tiene que pensar en seguir alguna de las tendencias ya establecidas, y por lo tanto, se tendrá que enfrentar a este tipo de preguntas.

1.3 El usuario actual

El usuario típico de una computadora es cualquier individuo, desde niños hasta ancianos, estudiantes, amas de casa y profesionistas. Crear software para este público definitivamente no es nada sencillo. Sin embargo, al usuario no le importa en lo más mínimo cuán difícil sea, él simplemente quiere obtener lo que espera de un sistema de cómputo.

1.3.1 Un ser exigente

Estamos viviendo los inicios de la era de la información. La vida de los seres humanos se ha visto transformada radicalmente en cuestión de una década. Internet ha cambiado nuestra manera de pensar y de relacionarnos entre nosotros y la tecnología móvil ha hecho de los seres humanos, seres constantemente comunicados. Parece ser que todo el tiempo nos encontramos sedientos de información.

Esta tecnología ha hecho que todo lo queramos hacer más rápido y mejor, queremos tener la información disponible en el momento exacto en el que queremos de ella y el no tenerla nos estresa³.

³Por supuesto que este es un fenómeno de los grandes centros urbanos.

Para el desarrollador, esta situación representa una presión constante. Siempre tiene que estar a la vanguardia, siempre tiene que darle al usuario lo que necesita en el momento que lo necesita. Sobre todo, el tan rápidamente cambiante mundo de las computadoras hace de los desarrolladores personas constantemente en movimiento. Pongamos un ejemplo.

En la década de los años ochenta la forma en la cual los niños se divertían cambió radicalmente. El surgimiento de las primeras consolas de videojuegos hicieron que los niños, desde muy temprana edad, desarrollaran habilidades muy serias en el manejo de sistemas de cómputo y equipo electrónico en general. Esta es la generación que actualmente está generando software.

Cada nueva consola de videojuegos, cada nuevo juego, tiene que tener por lo menos una diferencia con respecto al anterior: Tiene que ser superior en gráficos y sonido. Inclusive, al consumidor de esta tecnología no le importa mucho si la temática del juego se repite, simplemente tiene que ser superior tecnológicamente.

Esto obliga a los desarrolladores a no poderse quedar estancados, como ya dijimos, en casi ningún aspecto del software. Veamos otro terreno del software: El software de aplicaciones. En concreto pensemos en Microsoft Office. Prácticamente toda la funcionalidad presente en el actual Office XP existía ya desde la versión 97 ¿Cuál es la diferencia entre estas dos versiones? Básicamente se ven mejor y tienen ciertas características que al usuario le agradan. Si por alguna razón es más poderoso que antes, pero todo el aspecto visual es el mismo, el usuario sentirá una gran decepción.

Entonces el desarrollador tiene un problema más, se ve obligado a cambiar el programa no sólo en su funcionalidad sino también en su apariencia y reactividad⁴, y esto con cada nueva versión del software.

Tenemos entonces un usuario exigente ¿Cómo pueden los desarrolladores cumplir con todas estas exigencias? Con tantos cambios que hacer a la interfaz ¿Cómo pueden estar seguros de que mejoran el programa? ¿Cómo saber

⁴Con reactividad queremos expresar la forma en la cual el programa reacciona ante las entradas del usuario, esto incluye formas de animación y pequeñas reacciones del software que agraden al usuario, por mencionar algunas.

que no se están cometiendo errores? ¿Cómo saber que el usuario no esperaba tal cosa o no le gusta tal otra?

El único medio objetivo con el que contamos para validar los requerimientos del cliente son las evaluaciones con el usuario

Desde el punto de vista de esta tesis, estas evaluaciones son la herramienta más poderosa para darle al usuario el software que requiere.

1.3.2 El software desde el punto de vista del usuario

El software es una herramienta más dentro de todas las que posee el ser humano. Desde cierta perspectiva el software no es muy diferente a una licuadora, o un martillo. La diferencia radica en que mientras estas herramientas funcionan extendiendo las capacidades físicas del ser humano, el software es una herramienta que extiende la mente y esto lo hace especialmente delicado.

Muchas veces se menciona la palabra "amigable" cuando se habla de programas de cómputo. ¿Realmente el usuario quiere que la computadora sea amigable? o mas bien ve al software de manera no muy diferente a como ve una tostadora de pan.

Los aparatos que existen hoy en día presentan un nivel de funcionalidad que para el usuario es en general atractiva. Los teléfonos celulares actuales son también agendas, navegadores, reproductores mp3, calculadora y podemos seguir indefinidamente. Manejar toda esta funcionalidad es definitivamente muy difícil, sin embargo se espera que hacer una llamada telefónica sea igual de fácil que siempre lo ha sido. Esto es:

Un cierto sistema debe ser siempre tan fácil de usar como lo permita la actividad misma

Muchos sistemas de software, sin embargo presentan serias dificultades de uso a pesar de que la tarea para la cual están diseñados es sencilla. ¿Cuántas veces nos hemos frustrado con alguna funcionalidad básica de un programa que no podemos encontrar? Esto puede tener muchas causas pero al usuario eso no le importa. El no quiere usar un programa para tener problemas,

quiere usarlo para que mejore de alguna manera su vida.

Como ya se mencionó anteriormente, el usuario muchas veces no obtiene lo que espera. Los desarrolladores jamás se enteraron de que cierta pequeña funcionalidad sería de gran ayuda porque jamás hicieron pruebas reales con usuarios reales.

Así, en este trabajo, se pretende reformular una filosofía de desarrollo de software *centrado en el usuario*. Se pretende plantear que el usuario es importante y que no importa qué tan bien preparado esté el equipo de desarrollo, nunca podrán saber con certeza qué tan bien funcionará su programa hasta que no se hagan las pruebas correspondientes con usuarios reales. Esto además tiene que estar perfectamente bien detallado, dándole al equipo de desarrollo todas las herramientas para hacerlo.

1.4 La motivación

De entre todas las propuestas de procesos de desarrollo de software actuales, hay una especial, no sólo por ser nueva sino también por estar respaldada por experiencia de muchos años de investigación y porque pretende *unificar* la manera de crear software dentro de toda la IDS.

Esta propuesta lleva por nombre *Proceso Unificado de Desarrollo de Software* o simplemente Proceso Unificado (PU) [Jacobson *et al*, 2000]. El PU es uno de los temas principales de este trabajo y por lo tanto hablaremos mucho más a fondo de él a lo largo de todo el escrito.

Bien, las secciones anteriores nos dan lo que necesitamos para plantear las motivaciones del trabajo. Entre las ideas principales figuran las siguientes:

- Los productos de software están pensados para ser usados por los seres humanos.
- La IDS debe tener como una de sus principales prioridades al usuario.
- La IDS crea productos de software mediante procesos de desarrollo muy específicos.

- Los procesos de desarrollo actuales no contemplan de manera explícita al usuario. En particular, el PU da muy pocas ideas acerca de cómo lidiar con el problema de la usabilidad.
- El PU debe complementarse de manera que los desarrolladores tengan una buena guía acerca de cómo realizar un desarrollo centrado en el usuario.

Teniendo esto en cuenta, la idea motivadora es entonces la siguiente:

Un proceso completo de desarrollo de software debe contemplar todos los aspectos importantes que intervienen al crear buen software. Uno de estos aspectos es la importancia que tiene el usuario como consumidor del producto.

1.5 Los objetivos

Tomaremos al PU como proceso “blanco” de este trabajo. La razón es que presenta características que lo hacen especialmente innovador e interesante: Pretende unificar el proceso de desarrollo de software, su surgimiento está acompañado por un lenguaje formal y puede ser automatizado; además de que cuenta con el respaldo de las más importantes compañías de software a nivel mundial. Ahora pensemos en las características que debe de tener un DCU (desarrollo centrado en el usuario) y enfoquémonos en un aspecto específico del mismo para incorporarlo al PU.

Un DCU se compone de muchas etapas en las cuales se modela el producto, se diseña, se implementa y se prueba. La etapa que nos interesa es la última, la etapa de pruebas. Esta etapa es crucial en el desarrollo, ya que el usuario, al ser un ser humano, presenta algunas características que hacen difícil predecir su comportamiento. Las pruebas de usuario son la única manera de saber con cierto grado de certeza si lo que se ha contruido satisface las necesidades de usabilidad requeridas. Los objetivos de las pruebas de usuario pueden ser varios, dependiendo de la parte del proceso de desarrollo que se esté atacando:

- *Validar requerimientos.* Asegurar que el equipo de desarrollo identificó y plasmó de manera correcta las necesidades del usuario.

- *Validar diseño.* Asegurar que las necesidades de diseño capturadas por el equipo de desarrollo ayudan de manera efectiva al usuario en su tarea.
- *Validar desempeño.* Asegurar que el usuario puede realizar las tareas requeridas eficientemente.

Así, de lo que se trata es de ampliar algunas partes del PU de manera que los desarrolladores tomen en cuenta la importancia que la usabilidad y las pruebas de usuario poseen. Lo que se pretende es tener finalmente un proceso completo, un proceso que responda a preguntas como las siguientes:

- ¿En qué momento del proceso se requieren realizar las pruebas con usuarios?
- ¿Quién debe realizarlas?
- ¿Cómo deben de realizarse dentro del contexto del PU?
- ¿Qué tipo de información se obtiene y cómo debe de analizarse?
- ¿Cómo se inyectan estos resultados al proceso de desarrollo?

En el lenguaje del PU, contestar estas preguntas requerirá de la creación de trabajadores, actividades y artefactos para cada fase del proceso y cada flujo de trabajo, así como especificar el cómo, cuándo, dónde y por quién. Al final, se tendrá un mejor proceso, escrito totalmente en el lenguaje del PU y en el cual las pruebas de usuario sean parte integral del mismo.

Capítulo 2

Ingeniería de software actual

Para entender porqué las cosas son de tal manera actualmente, se tiene que entender su evolución desde los inicios. En este capítulo se hablará de la *ingeniería de software* (IS) desde el enfoque que nos interesa. Esto es, desde un enfoque de usuario y usabilidad ¹.

2.1 Breve historia de la Ingeniería de software

No es el objeto de la presente sección el simplemente repasar la historia de la IS². La IDS se ha enfrentado desde su inicio a múltiples desafíos. Así mismo ciertos acontecimientos la acercaron más a la solución de algunos de estos

Lo que nos interesa es cómo saber que se está creando software de calidad en términos de la usabilidad del mismo. Entonces, veremos a la historia de la IS desde este punto de vista y veremos cómo, poco a poco, la IDS ha podido ser capaz de hacer que los desarrolladores se concentren en aspectos de alto nivel de manera que la usabilidad del software y el interés que ponen los procesos en ella, es cada vez mayor.

¹Las principales fuentes de información para este capítulo fueron: [Endres, 1996], [Glass, 1997].

²Este tema y ha sido tratado ampliamente en otras tesis de ingeniería de software.

2.1.1 La era pionera

Esta era, comprende el período que va desde 1956 hasta 1967 aproximadamente. En ella, las primeras computadoras empezaron a tener una importancia significativa en la industria, las pocas computadoras que existían eran de tipo "batch" y el objetivo principal del desarrollo de software era optimizar los pocos recursos con los que contaban las máquinas. Los primeros compiladores como el FORTRAN 7090 demostraron que este mismo tipo de optimización podía también ser realizado por las máquinas.

El surgimiento del compilador marca la aparición de la primera *automatización* de la industria del software. Este fue un evento sumamente importante. Los programadores podían escribir código más complejo al manejar un lenguaje más cercano a ellos, además de que se podía reutilizar, mientras que la máquina se encargaba de realizar mucha de la labor mental que los humanos realizaban antes.

Durante la segunda mitad de la década de los sesentas, cambió el interés de los lenguajes a preguntas más generales acerca de las herramientas de desarrollo. El objetivo principal era reducir el tiempo necesario para ciertos aspectos de la codificación. Como ejemplo de tecnología de software inventada en esa época tenemos la compilación incremental y los *debuggers*. Se puede decir que el modo de operación de las computadoras de esta época era "interactivo", el dispositivo primario de comunicación con las mismas era un teclado.

Fue durante esta era que dos de los problemas principales en el desarrollo de software fueron identificados: "codificar y arreglar" y "código espagueti"³. Pero aún más importante, la gente empezó a darse cuenta de que había mucho más en el desarrollo de software que solamente codificar. De hecho, empezó a hacerse patente que los más altos riesgos en un proyecto de software estaban en las fases anteriores a las pruebas y la codificación. Esto motivó un nuevo enfoque y generó nuevos cambios.

³El primero se refiere a que los programadores codificaban, probaban si su programa corría y arreglaban los problemas, el segundo problema se refiere a que los programas no eran estructurados y por lo tanto eran sumamente difíciles de leer y mantener por programadores diferentes.

2.1.2 Dominando los procesos

La era de los procesos abarca aproximadamente de 1968 a 1981. Durante estas fechas, una industria de software independiente surgió, i.e., la industria creaba software independientemente de las tendencias en el hardware. También es común asociar esta era con la primera aceptación de una crisis en la IDS.

IBM introdujo la IBM 360⁴, la primera computadora capaz de tener múltiples funciones, podía combinar aplicaciones científicas y de negocio en ella misma y podía manejar lógica aritmética además de binaria. Por fin las personas que hacían software podían pasar más tiempo escribiendo código nuevo que actualizando el código viejo. Algunas veces el principio de esta era se asocia con el surgimiento de la IBM 360.

El objetivo principal perseguido en esta era fue reducir los riesgos y mejorar la calidad y la productividad así como identificar las causas de fallas en los proyectos de software. Se introdujo el modelo de desarrollo en cascada y se descubrió que la inspección de código⁵, si se hace bien, puede ser más efectiva que las pruebas exhaustivas para encontrar defectos en el código.

Para estos momentos, el concepto de proceso de desarrollo de software había surgido debido a la necesidad que se tuvo de manejar la creciente complejidad en los sistemas. Entonces, como se tenía ya todo un proceso a seguir para construir un sistema, el término de ingeniería de software empezó a ser acuñado.

Para mediados de la década de los setentas, la programación estructurada era ya común y la estabilidad de la disciplina dió surgimiento a disciplinas académicas relacionadas con el cómputo. El software empezó a tener un valor inmenso para las compañías.

Ciertamente, la mayor enseñanza de esta era fue que la calidad de un producto de software no puede ser medida tan sólo inspeccionando el producto final. La calidad es algo que se tiene que cuidar a lo largo de todo el proceso

⁴360 hacía alusión a que tenía una funcionalidad de 360°, i.e., podía correr aplicaciones de cualquier naturaleza

⁵Revisar el código creado por otro programador.

de desarrollo. Este fue otro gran avance, si vemos esta enseñanza desde una perspectiva más general, podremos ver que cualquier aspecto de la calidad tiene que tomarse en cuenta durante todo el proceso de desarrollo. Esto por supuesto, incluye a la calidad en términos de la usabilidad del sistema y será muy importante para este trabajo.

Una vez que se tuvo claro el camino a seguir en términos de los procesos necesarios, se empezó a perseguir el formalismo⁶ en la industria con un propósito, esto es, se quería lograr un mayor nivel de automatización de procesos. El objetivo perseguido por los métodos formales en software es incrementar la confiabilidad de software y mejorar la productividad por medio de la automatización. Las herramientas formales creadas fueron lenguajes de diseño y lenguajes de especificación.

Pronto se hizo patente que, para proyectos grandes, ni los requerimientos ni el diseño podían ser expresados en términos de un modelo matemático, especialmente los requerimientos no funcionales. Desafortunadamente sucede que, para cada especificación de requerimientos corresponde un conjunto de requerimientos no funcionales, i.e., requerimientos que no tienen directamente que ver con la funcionalidad requerida del programa, por ejemplo: Encontrar una combinación agradable de colores para la interfaz o generar efectos visuales eficaces que guíen al usuario en su tarea.

A pesar de esto, como veremos en los capítulos siguientes, se han encontrado maneras de formalizar la mayor parte de los requerimientos. Algunos aspectos de los mismos, sin embargo, siguen representando problemas para ser formalizados adecuadamente.

Como veremos en la siguiente subsección, la ingeniería de software ha avanzado mucho en algunos aspectos de la creación de software, pero existen otros que han persistido durante mucho tiempo. Tal vez, sin embargo, esté llegando el momento de atacarlos, ya con otras herramientas y filosofías de desarrollo. En concreto, trataremos de evidenciar algunas de las razones por las cuales pensamos que la IS está lista para tratar a la usabilidad más seriamente.

⁶Como ya se mencionó anteriormente, formalizar algo en nuestro contexto corresponderá simplemente a ponerlo en una forma que todos dentro de la disciplina lo entiendan.

2.1.3 Dominando la complejidad

La complejidad empezó a abundar en los sistemas creados a partir de 1982. En parte, gracias a los avances logrados en el terreno de los procesos en la era anterior, la complejidad en los programas, aunque aumentó grandemente, pudo ser manejada.

El gran evento en esta era fue indiscutiblemente el surgimiento de la PC: Por primera vez, los programadores podían tener computadoras en sus escritorios. Fue también en esta era cuando el dominio del hardware sobre el software terminó. El software empezó a dominar a la industria del cómputo.

Un evento que nos interesa de manera particular fue la proliferación de los monitores CRT (Cathodic Ray Tube). Esto trajo consigo la aparición de muchos dispositivos para acceder a la información en pantalla. En 1963 surge el "Sketchpad" de una tesis de doctorado, la primera aplicación gráfica de la historia. De pronto todo el mundo accedía a la información de una computadora por medio de monitores.

Ahora, lo realmente importante respecto a la proliferación del monitor fue que esto creó una nueva entidad en el software: La *Interfaz Gráfica de Usuario* (IGU). La idea se originó en el Xerox Palo Alto Research Center a finales de los sesentas y posteriormente fue hecha el núcleo del sistema operativo Machintosh. Actualmente es universal en cualquier sistema de cómputo pensado para ser usado por la gente.

Los grandes eventos interesantes para nosotros no iban a detenerse ahí. Así como el monitor posibilitó el surgimiento de las IGU, así también las IGU impulsaron de manera importante el surgimiento de las herramientas de tipo CASE, que es el acrónimo en inglés de *Computer Aided Software Engineering*. De la noche a la mañana la IDS fue capaz de alimentarse a sí misma, creando sus propias herramientas de software. Este fenómeno tiene mucho que ver con el hecho de que había más gente usando computadoras, pero pensemos que las herramientas CASE sin IGU serían muy limitadas.

La gran ventaja de las herramientas CASE, fue que los programas podían ser eventualmente mantenidos a nivel de diseño en vez de ser mantenidos a nivel de código fuente. Con estas herramientas, la IDS pensó que estaba en

camino de automatizarse a todo nivel. Sin embargo, la esperanza de que las herramientas CASE automatizaran la generación de código nunca se materializó del todo, por lo menos no en la primera mitad de esta era.

La orientación a objetos

El paradigma de programación conocido como *Programación orientada a objetos* (POO) ya existía desde los años sesenta con el primer lenguaje de este tipo: SIMULA I (1962-1965). Aunque fue SIMULA 67 el que realmente representaba el arquetipo de los lenguajes con *orientación a objetos* (OO).

Este nuevo paradigma de programación permitió que los programadores fueran capaces de manejar una complejidad aún mayor en sus programas. También permitió que el muy buscado reuso de código fuera mucho más fácil de lograr. La OO tiene el potencial para construir sistemas de software que tienen realmente una estructura modular y que son fácilmente divisibles en sub unidades, el reuso de código, por lo tanto, es más sencillo.

Similarmente a lo que ocurrió con métodos de programación estructurada, la OO fue primero aplicada a la codificación, después al diseño y posteriormente al análisis de requerimientos.

Con la OO, otro paso más hacia la automatización fue dado, pero además, esto traería beneficios para el aspecto de la usabilidad de los sistemas de software. La OO ya había permitido en los años ochenta la popularización de programas basados en IGU, esto es, programas a cuya funcionalidad se accede a través de una interfaz gráfica, sin embargo, las interfaces todavía tenían que ser programadas y éstas eran responsables en muchos casos de la mayoría del código total del programa, algunas veces superando el 75% del volumen total.

A principios de los años noventa, algunas compañías aprovecharon las bondades de la POO y sacaron al mercado entornos de programación (basados en *Lenguajes orientados a objetos* (LOO)) y que permitían generar muy rápidamente aplicaciones basadas en IGU que permitían interacciones complejas entre el usuario y el programa, i.e, se empezaba a automatizar la creación de IGU. Estas herramientas son llamadas herramientas RAD, el acrónimo en inglés de *Rapid Application Development*.

Actualmente, en unas horas, un pequeño grupo de programadores puede tener lista la totalidad de la IGU de un programa pequeño. Algo imposible de lograr antes de la aparición de los LOO y las herramientas RAD. Con el surgimiento de estas herramientas se acentuó la idea de que la construcción de la IGU era la parte fácil de construir un sistema de software. Con el tiempo sin embargo, se ha visto que el tener una IGU no garantiza la buena usabilidad de un sistema. Aún más, si bien la automatización de buena parte de la codificación por fin se ha logrado, la IDS todavía tiene retos de automatización, usabilidad y formalismo.

2.1.4 El presente intento de formalización

A mediados de los años noventa, la IDS presencié otro evento importante en la historia de la IS, el nacimiento del ya mencionado Proceso Unificado. El PU, reúne décadas de conocimiento en la investigación del desarrollo de software OO.

En el Capítulo 6 se hablará con más profundidad del PU, por ahora, mencionaremos las características que tiene y que hacen que tenga un lugar dentro de la historia de la IS.

El Proceso Unificado de Desarrollo de Software, o simplemente PU, es una propuesta más dentro de todas las propuestas que existen para crear software de calidad. Se aplica a los proyectos con desarrollo OO y propone un nuevo proceso de desarrollo, sus creadores proponen además un lenguaje gráfico para formalizarlo y claman que su automatización es posible. Entonces desde nuestra perspectiva, el PU resulta particularmente interesante, ya que por medio de éste podrían alcanzarse la automatización y el formalismo requeridos y, habiendo resuelto estas necesidades básicas, poder llevar nuestro interés más comodamente hacia la usabilidad.

2.2 El estado actual

Analicemos el estado actual del desarrollo de software desde la perspectiva que hemos venido manejando, la usabilidad, la cual se refiere al grado en el cual, un cierto software, es fácil de usar, es eficiente, se pueden realizar con

él las tareas en menos tiempo y el usuario se siente satisfecho.

Como se verá más adelante, la IDS todavía sigue considerando a la usabilidad y las pruebas de usuario como secundarias dentro del desarrollo de un sistema de software. Las razones han sido ya mencionadas, sin embargo, si ciertos avances prometedores como el PU tienen éxito, podríamos estar en posición de dar un paso más y empezar a resolver estos asuntos de usabilidad desde la raíz, esto es, desde el planteamiento del proceso de desarrollo mismo.

Actualmente tenemos programas sumamente complejos que muestran IGU llenas de diseño y animaciones atractivas. Así mismo, tenemos software que funciona con el mismo paradigma de interfaz gráfica que existe desde hace un par de décadas: Las ventanas, o el *point and click*. Mucha gente piensa que todo software que siga este paradigma es fácil de usar, incluso, algunos programadores y profesionales del cómputo en general, comparten esta idea.

Tomemos como analogía la aparición de los primeros procesadores de palabras en los años ochentas. De la noche a la mañana, la gente que siempre había escrito textos en una máquina de escribir sintió el poder de crear documentos libres de errores ortográficos y con una apariencia similar a la de un libro impreso. Cuando los procesadores de palabras *multifuente* y WYSI-WYG (What You See Is What You Get) aparecieron, todo el mundo abusaba de los diferentes tipos de letras disponibles. La gente no sabía que hacer con tantas herramientas de escritura. Sobra decir que aún con el procesador de palabras más poderoso se pueden escribir textos de la más baja calidad, tanto en contenido como en forma.

Igualmente que con los primeros procesadores de texto, los desarrolladores que probaron las primeras herramientas RAD sintieron que tenían el poder de siempre crear interfaces "amigables" y que la única preocupación era el *kernel* del programa. Sobra también decir que esto no es así, el crear una IGU de calidad requiere mucho más que un dominio sobre las herramientas RAD⁷

Más aún, los procesos de desarrollo de software actuales no contemplan como parte integral del proceso mismo, al usuario. Los asuntos de usabilidad

⁷Además, ni siquiera las herramientas RAD actuales proveen los medios para crear toda la complejidad gráfica requerida para ciertas aplicaciones.

quedan a criterio del equipo que hace el desarrollo.

El nuevo siglo está presenciando el surgimiento de una nueva generación de interfaces. Interfaces que, además de verse muy bien, se anticipan al usuario haciendo uso de técnicas de inteligencia artificial. Pero la insistencia sigue, todo esto debe de estar totalmente contemplado en el proceso mismo de desarrollo.

2.3 El objetivo

Con base en todo lo expuesto en las secciones anteriores y en el Capítulo 1, es fácil ver a dónde queremos llegar. Los ingenieros de software actuales tienen finalmente resueltos muchos de los problemas que los desarrolladores anteriores no tenían, pueden entonces enfocarse más en problemas de más alto nivel, en particular, en la usabilidad.

El PU es importante, y la usabilidad lo es aún más. Y la única manera con la cual se puede saber con certeza si una interfaz de usuario funciona como debe funcionar con usuarios reales, es por medio de pruebas de usuario. Por lo tanto, se pretende agregar al PU todo lo necesario para que los desarrolladores que escojan seguir este proceso tengan una guía completa acerca de cómo conducir estas pruebas de manera que se obtenga una interfaz de usuario de calidad.

Capítulo 3

Interacción humano-computadora

Ya se habló de diferentes aspectos de la IS y cómo se mezclan los objetivos de este trabajo en el contexto de su historia. Ya fue introducido también el PU y se habló un poco de aspectos de éste que lo hacen especialmente interesante para este trabajo. Existen, sin embargo, dos entidades de las cuales falta más información: El usuario y la Interacción Humano-Computadora (IHC).

Dedicaremos este capítulo a hablar de IHC, inevitablemente tocaremos aspectos muy importantes del usuario, aunque se tocarán aspectos diferentes a los que se pretende tocar en el Capítulo 4. En este capítulo importarán aquellos aspectos que estudia la IHC, i e, aquellos aspectos que nos hacen conocer al usuario y entenderlo, y aplicar esos conocimientos para mejorar su relación con la computadora. En el capítulo 4, en cambio, nos interesarán más aquellos aspectos del usuario que funcionan como motivadores de la IDS. Esta diferencia quedará clara más adelante.

3.1 Interacción humano-computadora

Actualmente no hay acuerdo en cuanto a la definición o el rango de tópicos que abarca el área conocida como *Interacción humano-computadora*. Sin embargo, para propósitos de enseñanza e investigación se tiene que dar una definición con la cual se pueda saber qué investigar y qué enseñar. Escogemos una definición sobre la cual nos basaremos a lo largo del presente trabajo

[Hewett *et al*, 1996]:¹

La interacción humano-computadora es una disciplina concerniente al diseño, evaluación e implementación de sistemas de cómputo interactivos para el uso humano y al estudio de los principales fenómenos que los rodean.

Desde una perspectiva de Ciencias de la computación, el enfoque es específicamente en la interacción entre uno o más humanos con una o más computadoras. Sin embargo, si variamos lo que se entiende por “humano” y lo que se entiende por “computadora” el alcance de IHC se hace claramente más amplio ¿Cuándo se puede decir que un fenómeno debe ser estudiado por IHC? Es simplemente cuestión de criterio.

Los sistemas *empotrados*² frecuentemente presentan interfaces gráficas al usuario, IHC entonces rebasa el concepto común de computadora y aborda los problemas relacionados con estos sistemas y su interacción con los usuarios. Este mismo problema de interacción con los sistemas empotrados también es abordado por una disciplina más general que IHC, se conoce como IHM (interacción humano-máquina) y claramente, si se relaja aún más el concepto de computadora podríamos estar hablando de cualquier sistema electrónico, electro-mecánico o aún puramente mecánico, lo cual nos pone ya fuera del terreno de IHC.

Pensemos también en que se puede relajar el lado humano, podemos pensar entonces en grupos humanos y organizaciones. IHC estudiaría en este caso la interacción de estas organizaciones con sistemas distribuidos. Si vamos más allá, hasta el terreno en el cual las computadoras raramente aparecen estaríamos una vez más fuera del alcance de IHC y bien dentro de otras áreas como pueden serlo la sociología

¹El documento al cual hacemos referencia aquí es el segundo capítulo del reporte generado por un comité escogido para establecer la definición, campo de estudio, perspectivas e importancia de la IHC así como para establecer el conocimiento mínimo que todo profesional de IHC debe tener.

²Los sistemas empotrados (*embedded systems*) son aquellos que incorporan sistemas de cómputo en máquinas generalmente no relacionadas con computadoras, refrigeradores, lavadoras, cafeteras o incluso tostadoras de pan pueden actualmente incorporar sistemas de cómputo con interfaces de usuario complejas.

El enfoque de IHC es diferente entonces al enfoque de las Ciencias de la computación. La diferencia más clara radica en que las Ciencias de la computación son estudiadas principalmente por computólogos, mientras que IHC es un área multidisciplinaria. Emerge como interés de muchas otras disciplinas, entre las cuales se encuentra de manera importante, por supuesto, las Ciencias de las computación.

A IHC le importan entonces aspectos como el desempeño de las tareas que desempeñan humanos y computadoras; la estructura de la comunicación entre el humano y la máquina; las capacidades humanas para utilizar computadoras; los algoritmos y la programación de las interfaces; los problemas ingenieriles que surgen en el diseño y la construcción de interfaces; los procesos de especificación, diseño e implementación de las mismas, por sólo mencionar algunos ejemplos.

Tomando en cuenta lo anterior, IHC puede ser de interés a disciplinas como las Ciencias de la computación, la Psicología, la Sociología y Antropología, el Diseño industrial, la Neurología y la Fisiología. Incluso algunos autores [Hewett *et al.*, 1996] establecen que IHC debe ser considerada una disciplina que pertenece igualmente a las Ciencias de la computación como a cualquier otra disciplina. Este último punto nos parece sin embargo discutible ya que puede ser que sean las ciencias de la computación las que más requieran de los conocimientos que genera IHC para su desarrollo.

3.2 Evolución e importancia de IHC

Las IGU deben ser programadas por ingenieros de software; Además, es conocido que estas interfaces, si bien representan en promedio 48% de las líneas de código totales, en algunas ocasiones pueden representar cerca del 100% de éstas [Hix *et al.*, 1993]. Es por tanto intrínsecamente necesario entender cómo tomar decisiones acerca de la funcionalidad que tendrá el sistema, cómo llevar esta funcionalidad (que generalmente tiene varias soluciones) al usuario, cómo construir el sistema y cómo probar su diseño. Esto es parte de lo que se trata este trabajo.

Esta disciplina que sólo hasta fechas recientes ha adquirido importancia entre los profesionales del cómputo, estudia a los humanos y las computado-

ras en comunicación. Por ello, ha surgido a partir de conocimiento generado por disciplinas de ambos lados, el lado humano y el lado de la máquina.

IHC surge como un campo de estudio que tiene sus raíces en la graficación por computadora, el diseño de sistemas operativos, la ergonomía, la ingeniería industrial y psicología cognitiva. La graficación por computadora surgió tempranamente en la historia de las computadoras con los primeros dispositivos CRT y las primeras plumas ópticas. Esto condujo al desarrollo de varias técnicas de interacción humano-computadora.

Muchas de estas técnicas ya se encontraban descritas en la tesis de doctorado de Sutherland en 1963 donde se describe el Sketchpad, que como se mencionó en el Capítulo 2, fue la primera aplicación gráfica de la historia

La investigación en sistemas operativos desarrolló técnicas para interconectar dispositivos de entrada/salida, adaptó el tiempo de respuesta de las computadoras al de los humanos, lo que permitió soportar sistemas operativos gráficos y animaciones en tiempo real.

La interacción de los humanos con las computadoras también era un tópico natural para la ergonomía, donde una extensión cognitiva al campo era necesaria resultando en la actual *ergonomía cognitiva e ingeniería cognitiva*.

Una parte de IHC que es importante entender es que, más que ninguna otra disciplina, es afectada principalmente por las fuerzas de la tecnología. Esto debido a que involucra transductores entre humanos y máquinas, y a que los humanos son muy sensibles a tiempos de respuesta. Así, las interfaces humanas son mucho más sensibles a la tecnología que la mayoría de otras áreas de las ciencias de la computación.

IHC es un campo que ahora más que nunca está en rápida evolución, tanto por los niveles del avance tecnológico actual como por la creciente importancia que esta disciplina está cobrando. Lo que hoy conocemos como sistemas ergonómicos no tendrá nada que ver con los sistemas ergonómicos que habrá en algunos años.

Podemos esperar entonces que IHC nos dará en el futuro cercano sistemas

con las siguientes características:

- *Abundancia en gráficos* Capacidades gráficas como procesamiento de imágenes, transformaciones gráficas y animación en tiempo real serán componentes usuales en cualquier sistema PDA (acrónimos en inglés de *personal digital assistant*.) o cualquier otro sistema de cómputo. Esto aumentará las capacidades de interacción entre estos sistemas y las computadoras
- *Sistemas altamente funcionales* Todos los sistemas tendrán una gran cantidad de funciones asociadas con ellos. Claramente, es en estos sistemas donde habrá más desarrollo de IHC, ya que serán sistemas pequeños, con poco espacio para desarrollar interfaces, y por lo tanto, éstas tendrán que ser sumamente avanzadas.
- *Multimedia* Cualquier sistema, incluyendo a los sistemas portables, incluirán imágenes, voz, sonidos, video, texto y cualquier otro tipo de datos con formato. Esto nos llevará a nuevos paradigmas de IGUs.
- *Pantallas grandes y delgadas* Estos dispositivos permitirán tener IGUs en sistemas que van desde estaciones de trabajo y PDAs hasta sistemas empotrados tales como refrigeradores y cafeteras.
- *Interfaces para toma de decisiones y trabajo colaborativo*. Serán desarrolladas IGUs que permitirán la interacción entre grupos de trabajo y en tiempo real. Lo cual posiblemente cambiará aún más nuestra manera de pensar.

3.3 IHC en software

Hasta aquí se ha hablado de la evolución y perspectivas que nos presenta IHC. Hablaremos ahora de un área de aplicación muy específica y sumamente importante en IHC: el software. Ahora, cuando hablamos de software hablamos también indirectamente del proceso de desarrollo, pero ¿Qué lugar tiene el proceso de desarrollo dentro de IHC? Para el comité encargado de realizar una revisión de esta disciplina y formular una definición más contemporánea de la misma, existen cinco aspectos interrelacionados de IHC que se consideran capitales para ser estudiados a fondo [Hewett *et al*, 1996]:

- La naturaleza de la IHC
- El uso y contexto de las computadoras.
- Las características humanas
- Arquitectura de interfaces y sistemas de cómputo
- El proceso de desarrollo.

Las IGUs e IHC están estrechamente relacionadas, y a este respecto sucede que las IGUs son implementadas siempre por ingenieros de software, si bien su diseño no siempre lo es.

Sin embargo, aunque son los ingenieros de software los que llevan a cabo estas interfaces gráficas, éstos son nada más parte de un grupo de desarrollo que en su mejor caso, es interdisciplinario³. En su peor caso es llevada a cabo y diseñada por un equipo de programadores que siguen algún proceso de desarrollo y que no tienen conocimiento alguno sobre IHC.

Si como se expuso en la sección anterior, los sistemas serán en extremo multifuncionales y estarán cargados de material multimedia, forzosamente se necesitarán de IGUs sumamente avanzadas y bien diseñadas, de manera que se pueda acceder a toda esta información de manera eficiente.

Por supuesto, este reto se presentará no solamente en sistemas móviles y empujados, sino también en el software de todo tipo de computadoras en general. Es por tanto de capital importancia incorporar todo el conocimiento que se ha generado en IHC a los procesos de desarrollo de software en general, ya que, finalmente, las IGUs son siempre software.

Lo importante aquí, y que ya ha sido expuesto, es que el crear una buena IGU es algo que debe ser cuidado en todo momento en el proceso de desarrollo, desde sus inicios hasta su compleción. Esto es algo con lo que los profesionales relacionados con IHC están de acuerdo y algo hacia donde se pretende llegar en este trabajo.

³Este equipo puede ir desde ergónomos hasta psicólogos e ingenieros de software.

Los criterios de diseño de interfaces que se desprenden de la investigación que se ha realizado en IHC, dictan como debe verse y comportarse una buena IGU. Sin embargo, saber cómo diseñarla con base en criterios previamente establecidos no lo es todo. Si suponemos que el equipo de desarrollo entiende IHC e incorpora ese conocimiento para crear una buena interfaz, aún así no se tiene todo el problema resuelto.

Si bien la ingeniería ya ha desarrollado maneras de construir tecnología y de probarla en ambientes simulados por computadora. La ingeniería de IGUs tiene forzosamente que probar con usuarios aquello que construye. En lo sucesivo trataremos cada vez más de enfocarnos en este aspecto del diseño y la construcción de interfaces. Esto es, las pruebas

Propondremos una manera de incorporar las pruebas de usuario al PU, usando el mismo lenguaje que es usado en éste, de manera que los desarrolladores no sientan que están usando un proceso diferente. Como objetivo secundario, acercaremos más al software que sigue el PU con IHC

Finalmente habremos de comentar que esto parece muy oportuno: En [Hewett *et al*, 1996] se establecen los elementos más importantes a tener en cuenta cuando se piensa en un proceso de desarrollo que tome en cuenta IHC. Más concretamente:

- *El proceso de diseño.* Donde explícitamente se menciona la importancia de modelar la tarea del usuario
- *Técnicas de implementación y herramientas.* Estas técnicas y herramientas van desde *toolkits* para implementación de interfaces hasta técnicas de organización.
- *Técnicas de evaluación.* Estas técnicas engloban toda forma útil para validar la interfaz creada, en concreto, pruebas de usuario.
- *Patrones de diseño.* Sugiere estudiar los diseños que ya han resultado ser efectivos.

Haremos caso al tercer punto y propondremos un proceso de diseño completo y esperamos, mejor. Los desarrolladores verán con más claridad cómo llevar a cabo las pruebas necesarias para asegurar la implementación de buenas IGUs.

Parte II

Pruebas de usuario e ingeniería de software

30

Capítulo 4

Ingeniería de software y usabilidad

En este capítulo hablaremos de IS y estudiaremos algunos procesos de desarrollo desde un punto de vista de la usabilidad del software. Analizaremos la meta de la IS y de los procesos de desarrollo e intentaremos mostrar que algo importante falta.

4.1 Sobre los procesos de desarrollo

¿Qué tenemos que hacer para decir que estamos haciendo IS? Por lo menos tenemos que construir software siguiendo un proceso estructurado de desarrollo. Si el proceso es un buen proceso y lo seguimos cuidadosamente, estaremos en camino de hacer software correctamente. Si además nuestro producto no tiene errores de codificación, es estable y hace lo que tiene que hacer, entonces, como buenos ingenieros de software diremos que hicimos un *buen* producto de software.

Ahora, podría ser que la curva de aprendizaje del sistema se extendiera demasiado y que fuera muy fácil cometer errores con él, aunque estas características no invalidan las del párrafo anterior ¿podríamos seguir asegurando que nuestro producto de software es bueno? ¿Qué tanto? ¿Cómo podemos asegurarnos de que construimos algo bueno?

Como veremos en el presente capítulo, los procesos de desarrollo de soft-

ware nos ayudan a construir un producto de software y a obtener información acerca de qué tan bien lo construimos y qué tan bueno es en términos de estabilidad, errores de codificación y satisfacción de requerimientos funcionales. Ahora, los sistemas de software tienen una característica que no es suficientemente contemplada por los procesos de desarrollo: La usabilidad

4.2 El usuario y la interfaz

La usabilidad surge de la relación entre el usuario y la máquina, una buena relación es siempre deseable en todo sistema de software. Como ya mencionamos en el primer capítulo, el usuario tiene muchas características que afectan la manera en la cual el software que fabricamos se desempeña. En primer lugar tenemos que recordar que:

Para el usuario, la interfaz ES el software

¿No es acaso esto suficientemente importante como para prestarle especial atención al diseño e implementación de la interfaz dentro del contexto del desarrollo mismo? Tanto el usuario como la interfaz son entidades que necesitan ser tratadas de diferente manera a como se tratan comúnmente otras entidades (código, documentación, recursos humanos) dentro del desarrollo.

Para diseñar algo complejo, generalmente se empieza por un modelo. Este modelo, no es la representación exacta de lo que vamos a construir, sin embargo, se encuentra suficientemente cerca como para poder realizar suposiciones y obtener verdades del objeto real. El modelado entonces nos permite entender cosas complejas. Bien, el software realmente lo es.

Ahora, el software resulta de la compilación de código ¿Cómo modelamos código? Si estamos hablando de código escrito en un lenguaje con orientación a objetos, podremos entonces utilizar UML [Jacobson *et al*, 1999]. De hecho, con UML se pueden modelar también flujos de trabajo, recursos humanos, actividades y muchos de los aspectos que juegan un papel importante en el desarrollo de software. Pero ¿cómo modelamos al usuario? ¿cómo modelamos la interfaz?

4.3 USUARIO E INTERFAZ EN LOS PROCESOS DE DESARROLLO 33

El usuario es el elemento más difícil de modelar. La interfaz, al ser construida por ingenieros de software, es un poco más accesible, sin embargo, el problema de modelar la IGU es un problema que no está completamente resuelto. Muchos libros de ingeniería de software sugieren modelar la IGU por medio de máquinas de estado finitas [Jacobson *et al*, 1999] [Hamlet, Maybee, 2001] [Hix *et al*, 1993]; bien, resulta que el modelo de una simple ventana con un par de botones y algunos campos de texto resulta en un diagrama complejo y de difícil lectura cuando se toma en cuenta la validación requerida para la información en los campos. Modelar grandes sistemas de software por medio de esta técnica es entonces problemático.

Aunque existen otras formas propuestas para modelar formalmente la interfaz (un ejemplo es UAN [Hartson *et al*, 1990]), no contamos con maneras eficientes de modelar al usuario y la interfaz. Sin embargo tenemos la necesidad de saber qué decisiones están bien y mal respecto a ambos. La respuesta a este problema son las pruebas.

Sólo las pruebas objetivas pueden, finalmente, resolver algunas áreas grises donde no es claro lo que funcionará y lo que no
[Constantine, 1995]

Entonces, a pesar de que tenemos dos aspectos muy importantes dentro del desarrollo de software que son difíciles de modelar, sí tenemos una herramienta poderosa que nos proporciona una solución alternativa al problema. Probando lo que contruimos, podemos llegar, iterativamente, a obtener algo de la calidad buscada a pesar de no contar con los modelos necesarios.

4.3 Usuario e interfaz en los procesos de desarrollo

En esta sección haremos una revisión a la manera con la cual se ataca la usabilidad en algunos procesos de desarrollo importantes. Esto nos dará un panorama del papel que juega la usabilidad en la IS actual.

4.3.1 El Proceso Unificado

Antes de revisar la usabilidad dentro el PU, daremos una breve introducción al mismo.

¿Qué es el Proceso Unificado?

El PU [Jacobson *et al*, 2000] es una metodología de desarrollo de software con OO que innova en la creación de software en varios aspectos. Propone un lenguaje gráfico y formal (UML) para modelar flujos de trabajo, actividades humanas y productos del desarrollo. Así mismo, mediante herramientas basadas en este lenguaje, la estructura PU permite automatizar ciertos aspectos del desarrollo¹, desde la creación de código hasta la creación de documentación formal [Rational Unified Process].

El PU es un proceso iterativo e incremental, esto significa que se crea un producto que, mediante iteraciones, se va acercando cada vez más al producto final. Estas iteraciones van desde la captura de requisitos hasta las pruebas, i.e., se crea el producto final a base de microproyectos.

El PU consiste en cinco flujos de trabajo y cuatro fases como lo muestra la Figura 4.1. Cualquier número de iteraciones pueden ocurrir en cada fase, sin embargo, como lo muestra la misma figura, no todos los flujos de trabajo ocurren con la misma intensidad a lo largo de todo el proceso: Mientras que en el inicio la captura de requisitos es muy intensa, cerca del final del proyecto ya no ocurre.

Este proceso es dirigido por casos de uso², éstos permiten realizar la captura de requisitos en el lenguaje del cliente y luego traducirlos en una representación formal de ellos en UML. Mediante los casos de uso se pulveriza la funcionalidad del sistema tanto como el desarrollador lo quiera.

El PU también es centrado en la arquitectura³: Es de interés al proceso saber cómo se relacionan las clases entre ellas así como la manera en la cual se mandan mensajes los diferentes objetos. Estos serían ejemplos de una vista estática y una vista dinámica respectivamente. Todas las posibles vistas sistema forman la arquitectura del sistema. La arquitectura de un sistema

¹Aunque la estructura de otros procesos de desarrollo también podrían aceptar herramientas que permitieran esto, el PU es el único proceso que hasta ahora ha demostrado la factibilidad de ser automatizado

²Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante.

³La arquitectura del sistema representa los aspectos estáticos y dinámicos más importantes del sistema.

4.3 USUARIO E INTERFAZ EN LOS PROCESOS DE DESARROLLO

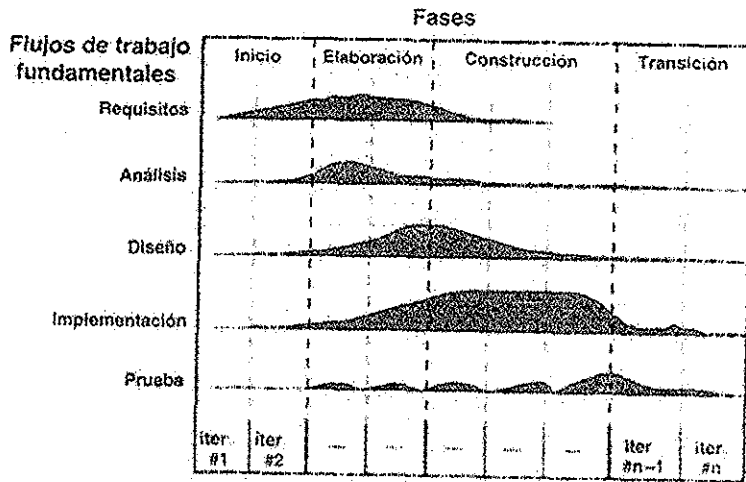


Figura 4.1: El Proceso Unificado itera flujos de trabajo a través de cuatro fases

tiene su representación en UML

UML

Una de las partes más importantes en el Proceso Unificado es el modelado. El modelado utiliza modelos para crear una representación simplificada de la realidad. *Un modelo es la simplificación de un sistema o un problema, misma que nos ayuda a entenderlo.* El Proceso Unificado propone la creación de modelos durante el desarrollo del proyecto. Así, durante el proyecto, se crea el modelo de casos de uso, el modelo de análisis, el modelo de diseño, el modelo de implementación y el modelo de prueba.

UML [Jacobson *et al*, 1999] es un lenguaje gráfico diseñado para modelar todos los aspectos involucrados en el proceso de desarrollo de software. UML pretende ser un estándar, lo cual inyectaría formalismo a toda la IDS. Como ya hemos mencionado anteriormente, este lenguaje posee diagramas especialmente diseñados para representar las vistas estáticas y dinámicas de un sistema, así como diagramas para modelar flujos de trabajo, trabajadores,

TESIS CON
FALLA DE ORIGEN

captura de requerimientos, actividades.

De esta forma, todo en un proceso de desarrollo puede ser representado en UML. Por otro lado, el PU clama ser una metodología completa de desarrollo. Ambos, el PU y UML dan como resultado una mancuerna con la cual se puede generar eficientemente software de calidad y de cualquier tamaño.

Proceso Unificado y Usabilidad

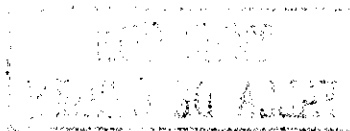
Habiendo hablado del PU, haremos una breve revisión de algunas partes de su contenido [Jacobson *et al*, 2000]⁴.

Por motivos de escasez, nos permitiremos cubrir todas y cada una de las menciones que se hacen dentro de este proceso al usuario, la IGU o en general, a la usabilidad.

- 1^a Se menciona que durante la fase de captura de requisitos, los analistas deben especificar cuál será la apariencia de la IGU. Se sugiere realizar varias versiones de prototipos para que los usuarios las prueben. Sección 6.3, "captura de requisitos funcionales".
- 2^a Se especifica la existencia del artefacto: *Prototipo de interfaz de usuario* (PIU). Su objetivo es ayudar a comprender mejor los casos de uso y especificar las interacciones entre actores humanos y el sistema durante la captura de requisitos. Nótese que se propone que el PIU se realice durante la captura de requisitos. Sección 7.2.6.
- 3^a Se introduce al *Diseñador de interfaz de usuario* (DIU), su trabajo es producir el artefacto mencionado en la referencia anterior. Sección 7.3.3.
- 4^a Se propone como actividad el prototipo de la interfaz. Para crear el diseño lógico de esta interfaz el diseñador tiene que traducir los casos de uso y los elementos del glosario de términos⁵ en elementos de la interfaz, i.e, hacer un mapeo de las palabras del glosario a elementos de interfaz. Al hacer esto se sugiere responder a algunas preguntas

⁴Esta revisión está basada en el texto oficial en español de este proceso de desarrollo

⁵El glosario de términos recoge todos los términos (palabras específicas) del negocio en el cual se utilizará el producto de software



4.3. USUARIO E INTERFAZ EN LOS PROCESOS DE DESARROLLO 37

tales como ¿Qué elementos de interfaz se necesitan para posibilitar los casos de uso? ¿Cómo deberían relacionarse unos con otros? ¿Cómo debe ser su apariencia? Estas preguntas las contesta el diseñador. Se proponen revisores que verifiquen la validez de las interfaces para que éstas puedan pasar a su diseño. Sección 7.4.4.

- 5^a Se introduce la *clase de interfaz*. Las clases de interfaz se utilizan para modelar la interacción entre el sistema y sus actores (usuarios). Generalmente, las clases de interfaz representan abstracciones de ventanas, formularios, paneles, interfaces de impresoras, sensores, etc. Estas clases siempre se deben mantener a un nivel bastante alto y conceptual y forman parte del modelo del análisis, mismo que se crea en el flujo de trabajo Análisis. Sección 8.4.2.1.
- 6^a En el flujo de diseño, durante la fase de inicio, puede optarse por desarrollar un prototipo de demostración. Este prototipo le será mostrado al usuario de manera que sea seguro que el mismo satisface sus necesidades. Cabe mencionar que esto se sugiere como una herramienta de convencimiento de los clientes. Sección 13.4.3.
- 7^a Durante el flujo de requerimientos, se *pueden* desarrollar prototipos de las interfaces, sólo si son interesantes desde un punto de vista funcional (de la arquitectura). Se comenta que existe una razón *adicional* para desarrollar prototipos de IGUs: Para averiguar si funciona utilizando usuarios reales. Sección 14.4.1.2.
- 8^a Se especifica que el prototipo de la interfaz de usuario debe realizarse hasta la fase de construcción. Esto es, una vez que el proyecto llegue a la fase de construcción, se empezarán a recabar los requisitos de interfaz del sistema. Se propone construir varios prototipos para que con base en pruebas de usuario se vayan refinando. Se indica que el prototipo de la interfaz se debe realizar en los requisitos y no en el diseño. Sección 15.4.1, "Desarrollar un prototipo de la interfaz de usuario".

Como podemos apreciar, las referencias a estas cuestiones, son en efecto, escasas. 13 páginas de un total de 405 para la versión en español. En el Capítulo 7 explicaremos más a fondo el trato que le da el PU al usuario, la usabilidad y las IGUs.

4.3.2 Extreme Programming

Extreme Programming (XP) es una metodología interesante de desarrollo de software que nació en 1996 de un proyecto para DaimlerChrysler y ha experimentado gran éxito en los proyectos en los cuales se ha aplicado. Forma parte de las metodologías conocidas como de *peso ligero* y, según sus creadores, es exitoso porque enfatiza la satisfacción del cliente. La referencia principal de este proceso de desarrollo se cubre completamente en poco tiempo [ExtremeProgramming.org, 2002]. Al ser una metodología sencilla, no requiere de cientos de páginas para su exposición.

XP, clama terminar los proyectos a tiempo, con bajo costo de producción, con código simple y corto y con un alto grado de satisfacción por parte del cliente e incluso del equipo de desarrollo. Está pensado especialmente para equipos de desarrollo de entre 2 y 10 programadores que trabajen en proyectos de alto riesgo y que presenten constantes cambios en los requerimientos.

La manera de lograr el éxito es mediante pruebas, se realizan pruebas antes, durante y después de la programación, de esta manera se logra eliminar la sobrevivencia de errores. Si un nuevo requerimiento surge, se incorpora inmediatamente al desarrollo y se generan nuevas pruebas. Las pruebas son lo más importante dentro de XP.

La entidad participante más importante dentro del proyecto es el cliente. El desarrollo arranca y el cliente es responsable de escribir las *historias de usuario*. Estas describen textualmente y en el lenguaje del cliente, lo que el sistema debe hacer.

A partir de las historias de usuario, se generan las *pruebas de aceptación*, las pruebas de aceptación validan si las historias de usuarios han sido correctamente implementadas. Una historia de usuario no se considera completa si no ha pasado su prueba de aceptación.

Bien, entonces el cliente queda siempre satisfecho, escribe lo que quiere, los desarrolladores lo implementan en una prueba de aceptación y se itera hasta que el resultado sea satisfactorio. La entidad más mencionada en la referencia principal de XP es el cliente. Las referencias a la usabilidad son nulas así como las referencias al usuario. Si bien las historias de usuarios son

4.3. USUARIO E INTERFAZ EN LOS PROCESOS DE DESARROLLO

algo muy importante, éstas están escritas por los clientes, los cuales generalmente no serán los usuarios del sistema. También, mientras que las pruebas son lo más importante para XP, éstas son de código y ¿cómo sabemos si construimos una buena IGU?

Un aspecto interesante de XP es que aceptaría muy fácilmente conceptos de DCU. Por ejemplo, XP sugiere crear *soluciones pico*, éstas serán programas simples destinados a explorar soluciones potenciales, programas que sólo cubran los problemas siendo analizados. Claramente, muchas de estas soluciones pico podrían ser prototipos de IGUs. Adicionalmente, las historias de usuarios pueden ser también escritas por los futuros usuarios del sistema, los cuales tienen una mejor idea de los requerimientos de usabilidad del mismo.

4.3.3 Team Software Process

Team Software Process es una metodología de desarrollo para grandes proyectos de software, es considerada de *peso pesado* y está diseñada para equipos de hasta 20 ingenieros de software que trabajan en proyectos que pueden incluso durar años para su compleción.

En esta subsección, recogeremos algunas ideas extraídas del libro *Introduction to the Team Software Process* [Humphrey, 2000], mismo que presenta una versión reducida de TSP llamada TSPi que sirve como introducción a TSP. Las ideas que presenta TSPi sin embargo, se mantienen en TSP.

TSP enfatiza el trabajo en equipo, la idea es que los proyectos de software generalmente fracasan debido a problemas en el equipo de trabajo y no debido a problemas técnicos. Un buen seguimiento de TSP garantiza entonces que los proyectos no fracasen y que se entreguen productos de calidad. Al igual que los dos procesos antes mencionados, TSP se basa en desarrollos iterativos, en los cuales, se va logrando un producto cada vez más completo. Mencionaremos tres ideas importantes (para nosotros) dentro de TSPi.

Primero, al inicio del desarrollo, el equipo tiene que realizar la captura de requisitos. TSPi sugiere que además de capturar los requisitos del cliente, se tenga contacto con los usuarios para obtener requisitos que los clientes muchas veces no son capaces de proporcionar.

Segundo, después de tener (o creer tener) los requerimientos completos, los desarrolladores tienen que realizar un diseño conceptual del producto a ser construido, para lo cual se tienen que hacer cuatro preguntas:

1. Basado en lo que sé ahora ¿cómo debo construir este producto?
2. ¿Cuáles son los componentes principales que se necesitarán para construirlo?
3. ¿Qué funciones deben proveer esos componentes?
4. ¿Cuán grandes, creo yo, deben ser estos componentes?

Como podemos apreciar, todas las preguntas para realizar el diseño conceptual están enfocadas en el aspecto funcional del producto, el código, los patrones de diseño y los componentes ¿En qué momento se preguntan los desarrolladores por las características de usabilidad del sistema? Podría ser en la tercera pregunta pero la manera en la cual se expone podría dar a entender que tales funciones son a nivel de código y no a nivel de uso, cuando en realidad en la pregunta 3 caben ambas.

Tercero, los desarrolladores tienen que plantearse en algún momento los estándares de diseño a ser utilizados TSPi menciona seis:

1. Las convenciones de nombrado.
2. Los formatos de interfaces (entre componentes del sistema).
3. Los mensajes de error y del sistema.
4. Los estándares de defectos.
5. El conteo de líneas de código.
6. Los estándares de representación del diseño.

Respecto al último punto cabría preguntarse ¿Acaso no hay estándares de diseño importantes dentro de la usabilidad de un sistema?

4.3 USUARIO E INTERFAZ EN LOS PROCESOS DE DESARROLLO41

Finalmente TSPi menciona la usabilidad, nueve líneas de texto, mismas que resumimos aquí:

La usabilidad ... requiere un libro entero ... es importante pensar en ella durante el diseño. Una manera de hacer productos usables es producir escenarios para cada función clave del usuario ... analiza estos escenarios y asegúrate de que reflejen el tipo de sistema que crees los usuarios necesitarán ... Si logras herramientas de prototipo eficientes, es una buena idea prototipar y demostrar todas las interfaces de usuario

Este texto deja a criterio del desarrollador el diseño de las IGUs, sugiere probar las interfaces con usuarios sólo en el caso de que los desarrolladores puedan crear prototipos rápidamente

4.3.4 Otras referencias

En *The Engineering of Software* [Hamlet, Maybee, 2001], se sugiere modelar la IGU por medio de una máquina de estado finita. El plan de pruebas para la IGU consiste entonces en verificar que el modelado haya sido correcto, recorriendo el diagrama de estados de una manera completa buscando inconsistencias y áreas de conflicto. Al respecto podemos decir que *una interfaz bien construida no es necesariamente una buena interfaz*.

La organización ISO (International Organization for Standardization) contempla y apoya varios estándares relacionados con usabilidad, IGUs y DCU [Bevan, 2001] Sin embargo, al ser documentos separados de los documentos que especifican los procesos de desarrollo, pueden pasar inadvertidos por el equipo de desarrollo. La Figura 4.2 muestra los estándares relacionados con usabilidad, IGUs y DCU. Mucha de la información contenida en estos estándares debe ser incorporada (cuando menos modestamente) a los procesos de desarrollo.

| | Principios y recomendaciones | Especificaciones |
|---|--|--|
| Calidad de uso dependiendo del contexto de uso. | ISO/IEC 9126-1: Ingeniería de Software – Calidad del producto – Parte 1: Calidad del modelo. ISO/IEC TR 9126-4: Ingeniería de Software – Calidad del producto - Parte 4: Métricas de calidad de uso. ISO 9241-11: Guías en usabilidad. | ISO 20282: Usabilidad en productos de uso diario. |
| Interfaces e interacción. | ISO/IEC TR 9126-2: Ingeniería de Software – Calidad del producto – Parte 2: Métricas externas. ISO/IEC TR 9126-3: Ingeniería de Software – Calidad del Producto – Part 3: Métricas internas. ISO 9241: Requerimientos ergonómicos para el trabajo de oficina con terminales visuales Partes 10-17. ISO 11064: Diseño ergonómico de centros de control. ISO 14915: Ergonomía de Software para interfaces multimedia IEC TR 61997: Guías de estilo para interfaces de usuario en equipos multimedia para propósito general. | ISO 9241: Requerimientos ergonómicos para el trabajo de oficina con terminales visuales. Partes 3-9. ISO/IEC 10741-1: Diálogo – Control del cursos para edición de texto. ISO/IEC 11581: Iconos, símbolos y funciones de los mismos. ISO 13406: Requerimientos ergonómicos para el trabajo con pantallas planas. ISO/IEC 14754: Interfaces basadas en plumas – Edición de texto en sistemas basados en plumas. ISO/IEC 18021: Tecnología de información – IGUs para herramientas móviles ISO/IEC 18789: Requerimientos ergonómicos y técnicas de medición para pantallas electrónicas. |
| Proceso de desarrollo. | ISO 13407: Procesos de diseño centrados en el usuario para sistemas interactivos. ISO TR 16982: Métodos de usabilidad de soporte para el diseño centrado en el usuario. | ISO/IEC 14598: Tecnología de información – Evaluación de productos de software. |
| Otros | ISO TR 18529: Ergonomía de la interacción humano-sistema – Descripciones del ciclo de vida de procesos centrados en el usuario. ISO 10075-1: Principios ergonómicos relacionados con la carga mental – Términos generales y definiciones. ISO DTS 16071: Guía para el nivel de acceso para interfaces humanas. | |

Figura 4.2: Estándares relacionados con usabilidad en sistemas informáticos.

4.4 La necesidad de evaluar interfaces dentro de los procesos

Parte de la información contenida en el libro del PU es, desde el punto de vista de IHC, incompleta. En la primera referencia a la usabilidad (ver 4.3 1), se especifica que los prototipos de interfaz deben realizarse durante la captura de requisitos. Sin embargo en la quinta referencia se explica que esto debe ser hasta que se llegue a la fase de construcción, misma que sigue a la fase de inicio y a la de elaboración. En el Capítulo 8, propondremos que la elaboración de los primeros prototipos, aunque muy vagos ocurra antes, esto es, de ser posible, desde la fase de inicio del proyecto.

¿Qué podemos decir de XP? Tenemos que XP, aunque es flexible y puede ser fácilmente adaptado, no contempla al usuario ni la usabilidad del sistema de la manera requerida. La referencia principal para XP se enfoca exclusivamente en las necesidades del cliente y no de los usuarios. Aquí hay otra cosa importante que debemos tomar siempre en cuenta: El cliente no siempre sabe lo que quiere. Un proceso que se enfoca en la satisfacción del cliente ciertamente generará satisfacción por parte de éste pero corre el riesgo de pensar en los usuarios como algo secundario, dejando pasar problemas importantes de usabilidad.

TSP presta mucha atención a aspectos de calidad como es la calidad del código escrito, pero deja a criterio de los desarrolladores la usabilidad del sistema. El equipo de desarrollo puede cumplir con la calidad tal y como la ve TSP y aún así generar un producto deficiente debido a su complejidad de uso.

Con base en lo expuesto en este capítulo queda claro que la IS (o por lo menos un sector de ella) sí contempla a la usabilidad como algo importante. Sin embargo, si los conceptos de usabilidad no se incorporan a los procesos, los equipos de desarrollo podrán ignorarlos fácilmente.

Algunos de los estándares mostrados en la Figura 4.2, contienen información especialmente útil al tema de esta tesis. El ISO/IEC 9126-1 por ejemplo, describe seis categorías importantes para la calidad del software durante su desarrollo, funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibili-

dad y portabilidad. La cuarta parte del estándar, el ISO/IEC 9126-4⁶ tiene que ver con métricas para medir la usabilidad. Pero, ¿Cómo se logra?

Ningún proceso comprensible o unificado, liviano o ágil está completo sin una forma simple, sistemática y efectiva de aumentar la usabilidad y diseñar la IGU para lograr las necesidades reales de los usuarios [Constantine et al, 1999]

Las pruebas de usabilidad con usuarios son precisamente una herramienta sistemática y poderosa que asegura la usabilidad de un sistema de software si se aplican correctamente. Adicionalmente ofrecen una alternativa efectiva para atacar algunos problemas que el modelado de interfaces resuelve deficientemente.

En este capítulo veremos que, además de ayudar y guiar a los desarrolladores a construir el producto, algunos procesos incorporan metodologías para ayudar a medir la calidad del mismo, sin embargo, hay aspectos fundamentales que contribuyen a la calidad de un producto de software que no pueden ser cuantificados por medio de los métodos propuestos en estos procesos.

Dado lo anterior, cabría preguntar si los procesos de desarrollo realmente están cumpliendo con las metas para las cuales fueron diseñados: *Asegurar que software de calidad se fabrique correctamente.*

En el Capítulo 7 propondremos una manera en la cual el PU puede contener las pruebas de usuario sin perder su esencia. Es ahora tiempo de hablar de las *pruebas de usabilidad con usuarios.*

⁶A la fecha de escritura de este trabajo, el ISO 9126-1 todavía no se encontraba en la fase de publicación.

Capítulo 5

Pruebas de usuario

En el capítulo anterior hablamos de IS y su relación con la usabilidad. Hicimos notar también que es importante que las pruebas de usabilidad se contemplen dentro y a lo largo de los procesos de desarrollo mismos. Ya que este trabajo pretende ser autocontenido, ha llegado el momento de hablar de las pruebas de usuario

5.1 Breve introducción al Desarrollo Centrado en el Usuario

Como se ha mencionado anteriormente, *usabilidad* se refiere al grado en el cual un sistema es fácil de usar. Ahora, es importante señalar que las pruebas con usuarios son la base de algo que ya hemos mencionado, el Desarrollo Centrado en el Usuario ó DCU. Definiremos DCU como [Rubin, 1994]:

Las técnicas, procesos, métodos y procedimientos para diseñar productos usables

Hablaremos un poco más sobre DCU de manera que podamos adentrarnos más en las pruebas conociendo un poco sobre el contexto en el cual se aplican. El DCU tiene tres principios básicos [Rubin, 1994]:

1. *Enfoque temprano en usuarios y tareas* Normalmente los equipos de desarrollo empiezan los proyectos muy preocupados sobre las características que deberá tener el sistema. Para lograr un DCU, se deberá

prestar igual atención a los usuarios que utilizarán el sistema e igual atención a la tarea que realizarán con él. En este sentido es recomendable que la parte del equipo de desarrollo que tendrá contacto con los usuarios tenga algún tipo de entrenamiento para saber cómo llegar a conocerlos y como modelar su tarea para entenderla y poder crear un sistema que tenga un diálogo eficiente hombre-sistema

2. *Mediciones empíricas de uso del producto* El equipo de desarrollo tiene que tener una manera de medir la facilidad de uso y la facilidad de aprendizaje del producto. Esto tiene que hacerse muy temprano en el proceso de diseño a través del desarrollo de prototipos de pruebas para usarse con usuarios reales.
3. *Diseño iterativo* El DCU pone énfasis en que el diseño iterativo de interfaces no se trata solamente de pequeños ajustes al final del desarrollo. El diseño iterativo es algo que se tiene que cuidar durante todo el proceso. En esencia, el verdadero diseño iterativo permite darle forma al producto a través de un proceso de diseño, prueba, re-diseño y re-prueba.

Las empresas que practican el DCU tienen características en común:

- Equipos de desarrollo multidisciplinarios.
- Desarrollo por fases y retroalimentación con los usuarios en todo momento.
- Una gerencia conciente de la usabilidad
- Una perspectiva de “aprende mientras construyes”.
- Metas y objetivos que incluyen a la usabilidad entre sus prioridades.

Bien, entonces podemos seguir con la pregunta ¿Qué es una prueba de usuario?

5.2 ¿Qué es una prueba de usuario?

Una prueba de usuario es un tipo de prueba más dentro de todos los tipos de pruebas que un equipo de desarrollo puede llevar a cabo para validar y

verificar el producto que construye.

En particular una prueba de usuario utiliza usuarios tipo, i.e., usuarios similares (de no ser posible los futuros usuarios mismos) a los que se espera usarán el producto, para recolectar información importante. A los usuarios se les irán presentando prototipos cada vez más completos. La información que se obtenga puede estar en forma de opiniones, críticas e impresiones acerca del prototipo o acciones provenientes de un uso controlado y guiado de todo el prototipo o una parte de éste.

Pongamos un ejemplo de una prueba de usabilidad clásica en donde se requieren los siguientes elementos:

- Un usuario tipo que probará el producto
- Una persona que lo guiará y le pedirá realizar ciertas acciones con el prototipo.
- Un prototipo funcional, i.e., uno que sea capaz de retroalimentar al usuario.
- La documentación adecuada, la cual puede incluir manuales, fichas con información útil al usuario, etc.

Al usuario se le da la bienvenida, se le informa acerca de la naturaleza de la prueba y del sistema, se le pide que responda algunas cuestiones acerca del prototipo y se le solicita que realice actividades específicas con el mismo. Estas actividades están encaminadas a contestar ciertas dudas que los desarrolladores tienen acerca del prototipo:

- ¿Se entiende la navegación?
- ¿Se entienden las etiquetas?
- ¿Son intuitivas las tabulaciones de los campos?

Mientras que el usuario lleva a cabo las actividades que se le piden, los observadores de la prueba (los diseñadores del prototipo) observan y graban la prueba misma, la existencia de estos observadores será conocida por el usuario *a posteriori* con el fin de simular lo más posible el ambiente de trabajo real.

Posteriormente el equipo de desarrollo llevará a cabo la tarea de analizar la prueba (utilizando el registro generado durante la misma) para generar la documentación que servirá para mejorar el prototipo en el siguiente ciclo o iteración del desarrollo.

Entonces, una prueba de usuario es simplemente una herramienta más en el desarrollo de sistemas, mismos que tendrán alguna clase de interacción con usuarios. Si se requiere de una definición más formal podremos decir que [Rubin, 1994]:

Una prueba de usuario es un proceso que emplea participantes (quienes pertenecen al grupo representativo para el cual está diseñado el producto) para evaluar el grado en el cual un producto cumple con ciertos criterios de usabilidad

5.3 La importancia de las pruebas de usabilidad

Pensemos en cualquier proyecto de ingeniería civil, un puente por ejemplo. El equipo de desarrollo, habiendo entendido completamente los requisitos del mismo podrá pasar a la etapa de diseño y posteriormente de construcción. En ésta, se aplican intensivamente matemáticas en forma de fórmulas probadas para determinar cómo deberá ser construido. A pesar de que los ingenieros saben que su diseño es correcto, realizan simulaciones en computadora y modelos a escala para corroborarlo ya que, como ellos bien saben, una cosa es el papel y otra es la realidad.

En el caso anterior, el producto a construir fue uno sujeto a las fuerzas de la naturaleza, fuerzas conocidas y un tanto predecibles. Pensemos ahora que como ingenieros de software vamos a construir una IGU. No tendremos matemáticas que modelen el grado de usabilidad de la misma, tendremos que confiar en nuestra experiencia, sentido común y ciertos criterios empíricos para determinar lo que es correcto y lo que no.

Aún así ¿quién o qué garantiza que las decisiones tomadas por los desarrolladores serán las correctas para la mayoría de los usuarios del futuro sistema? La respuesta es muy sencilla: No hay manera de garantizar esto. Es claro que cuando se diseña algo que tiene que ser usado por seres humanos

la solución no es una.

Hay muchos métodos para apoyar la usabilidad en el desarrollo de sistemas de software [Maguire, 2001], en particular, las pruebas de usuario nos permiten, iterativamente, construir una interfaz que converja a la satisfacción de los requerimientos de usabilidad de un sistema de software

5.3.1 La importancia para el cliente

Primero que nada debemos aclarar quién es el cliente: El cliente es el que ordena la construcción del software, ya sea una persona, una institución o una empresa.

El equipo que implementará el sistema puede seguir cualquier proceso de desarrollo y éste puede o no estar centrado en el usuario. Ahora, si comparamos el costo de hacer un sistema de software con y sin enfoque en el usuario, entonces, dependiendo de la complejidad del mismo, la diferencia puede ser importante para el presupuesto del cliente.

Pensemos en el producto terminado con y sin pruebas de usuario. Seguramente el sistema creado utilizando las pruebas de usuario como herramientas de desarrollo, será más caro. Existe la probabilidad de que ambos sistemas estén bien diseñados y sean fáciles de usar, esto obviamente tendrá que ver con la experiencia y la madurez del equipo de desarrollo.

Sin embargo la experiencia ha demostrado que las probabilidades de lograr un mejor software en términos de usabilidad son mayores si se aplican técnicas de IHC durante el proceso de desarrollo [Nielsen, 1993]. En algunos casos, una simple ventana bien diseñada ha representado ahorros de cientos de miles de dólares a empresas [Nielsen, 1993]. Entre las herramientas más poderosas con que cuenta el DCU se encuentran las pruebas de usuario.

Entonces, es cierto que el cliente pagará un desarrollo más caro pero obtendrá también beneficios que finalmente representarán un ahorro en otros aspectos de la adquisición y adopción de un nuevo sistema de software.

En primer lugar, como las pruebas de software permiten a los desarrolladores construir un software más fácil de usar, el cliente gastará menos

en capacitación. Evidentemente, si sólo se actualiza software antiguo, el personal podría no necesitar capacitación, ya que las metodologías serían esencialmente las mismas.

Sin embargo, pensemos en el caso en el que el cliente es una empresa. Cuando el sistema es nuevo o cambia mayormente, se requiere de un proceso de capacitación de todo el personal para el aprendizaje del nuevo software. Claramente, entre más grande sea la empresa más tiempo y dinero tendrá que invertir en capacitación. En caso de que los clientes sean el público en general (el cliente coincide con el usuario), éstos obtendrán un producto más fácil de usar y podrán ser más eficientes con éste.

Adicionalmente, se cometerán menos errores con el sistema, aumentará la productividad y los empleados se sentirán más contentos al interactuar con el programa. Entonces, a largo y corto plazo existen beneficios serios para el cliente, si bien, en un principio se sienta reacio a pagar un mejor desarrollo. Un cliente que entienda bien estos beneficios podrá ver que, para una empresa grande y a largo plazo, el ahorro puede ser muy grande.

En resumen, las principales motivaciones que se tienen para pagar un DCU, con buenas pruebas de usabilidad son las siguientes:

- Ahorro en capacitación del personal que usará el sistema.
- Menor tasa de errores en el uso del programa.
- Mayor productividad.
- Mayor satisfacción y mejor ambiente de trabajo.

5.3.2 La importancia para el desarrollador

Las motivaciones para el desarrollador también son importantes. Pensemos en dos escenarios: a) Se desarrolla un producto de software para su venta al público, b) Se desarrolla especialmente para una empresa o institución.

En ambos escenarios, uno de los beneficios más importantes para el desarrollador se desprende directamente de los beneficios del cliente: Si el cliente

está satisfecho, el cliente regresará con el desarrollador. Además, en el escenario a), si el cliente está contento con el producto, lo recomendará, generando clientes adicionales.

La aplicación oportuna de pruebas de usabilidad pueden ahorrar también tiempo de desarrollo: Al ir el diseño de las mismas en la dirección correcta desde el principio, se evitan iteraciones de corrección de interfaces tarde en el desarrollo. Consecuentemente, las fechas de entrega pueden ser calculadas con mayor precisión. El desarrollador entonces:

- Aumentará su prestigio con el cliente al crear sistemas de software más fáciles de usar y por lo tanto más eficientes. Se generan clientes potenciales.
- Evitará correcciones iterativas de interfaces, que pueden implicar correcciones de código o aún de requerimientos, haciendo más probable, potencialmente, que la entrega del sistema se realice a tiempo.

5.3.3 La importancia para el usuario

Llegamos finalmente al tercer participante dentro de lo que es el desarrollo de sistemas de software: El usuario, aquel que finalmente hará un uso cotidiano del producto y que se verá más directamente impactado por sus deficiencias.

Si las pruebas con usuarios se llevaron correctamente, el usuario final podrá esperar un software más fácil de usar. Al tener la curva de aprendizaje más pendiente, el usuario ahorrará tiempo de aprendizaje y podrá dominar más rápidamente al programa. Familiarizarse entonces con el nuevo sistema o con la nueva versión del mismo será más sencillo.

Si el sistema es fácil de usar y más gratificante¹ se podrán realizar las tareas más eficientemente con éste y la satisfacción del usuario se verá aumentada.

Si lo anterior ocurre con el usuario, los otros dos participantes dentro del proceso de desarrollo se verán también beneficiados como ya explicamos.

¹Pensaremos que un sistema gratificante es aquel que devuelve con resultados el tiempo invertido al trabajar con él. Un software poco gratificante, termina frustrando al usuario.

El círculo finalmente se cerrará y todos estarán más satisfechos y habrán sacado más provecho del producto, directa o indirectamente. Resumiendo, con la ayuda de las pruebas de usabilidad bien aplicadas a lo largo de todo el proceso de desarrollo, el usuario:

- Dominará más rápidamente el producto.
- Trabajaré más eficientemente
- Quedará más satisfecho con el sistema.

5.4 ¿Cómo llevar a cabo una prueba de usuario?

Si el equipo de desarrollo y el cliente finalmente se convencen de la necesidad de un DCU, quedará llevarlo a cabo. Ahora, un DCU completo contempla analizar la tarea del usuario y modelarla adecuadamente además de realizar las pruebas con los usuarios mismos. Sin embargo, con sólo llevar a cabo pruebas de usabilidad como las descritas anteriormente, el equipo de desarrollo puede aumentar la usabilidad de su producto sistemáticamente y a lo largo de todo el desarrollo para entregar un producto mejor.

Al inicio del presente capítulo dimos la definición de una prueba de usuario. En las siguientes subsecciones hablaremos más de las pruebas de usuario, explicaremos cómo prepararlas, cómo llevarlas a cabo y qué hacer con los resultados que de éstas surjan.

A partir de aquí y hasta la terminación de este capítulo, daremos información suficiente para que cualquier grupo de desarrollo pueda, si así lo desea, conducir una prueba de usuario para un sistema real y obtener de ahí resultados útiles que mejoren el sistema en construcción.

5.4.1 Las pruebas de usabilidad, los distintos tipos

Pensemos en un desarrollo basado en ciclos o iteraciones donde cada una de estas iteraciones equivale a un “microproyecto”. Al cabo de cada iteración se debe tener un producto terminado, el cual se acercará más con cada iteración

posterior, al producto final.

Bien ¿cómo entran las pruebas en este esquema? Las pruebas con usuarios se deben llevar a cabo al final (o cerca del final) de cada iteración. Estas pruebas arrojarán información que será utilizada en el inicio de una nueva iteración. Una prueba dada, por ejemplo, puede hacer del conocimiento de los desarrolladores que se requiere cierto elemento de la interfaz para poder funcionar como se tiene planeado. El equipo entonces tiene un nuevo requerimiento para la interfaz y el sistema en general. Se deberá arrancar la nueva iteración adicionando este nuevo requerimiento a los que ya se tienen para la nueva implementación. Al final, por supuesto esto podrá ser nuevamente probado con usuarios.

La frecuencia con la cual se deben de realizar estas pruebas con usuarios queda a criterio de los desarrolladores. Las pruebas de este tipo, al ser generalmente caras, pueden obligar a los desarrolladores a realizarlas sólo cuando se haya acumulado una cantidad considerable de dudas o cuando se tenga avanzada una buena parte del sistema.

Identificaremos a continuación tres tipos de pruebas que serán de interés en un desarrollo de software: Las pruebas exploratorias, las pruebas de diseño y las pruebas de desempeño. Debemos tener en mente que cada una de éstas puede ocurrir varias veces a lo largo de todo el desarrollo, tantas como el equipo de desarrollo lo considere necesario. Los objetivos de las siguientes pruebas ya fueron mencionados al final del primer capítulo, las analizaremos ahora más detenidamente.

5.4.2 El prototipo

No se necesitará evaluar el prototipo completo (que puede no existir). Se requiere sólo la pequeña parte de la funcionalidad que se quiere probar.

Se tendrá que decidir entre una prueba “horizontal”, una prueba “vertical” o una mezcla de ambas. Cuando se tiene un prototipo de un sistema, normalmente se tiene un árbol de pantallas o uno de ventanas según sea el caso.

Imaginémonos que tenemos un árbol como se muestra en la Figura 5.1,

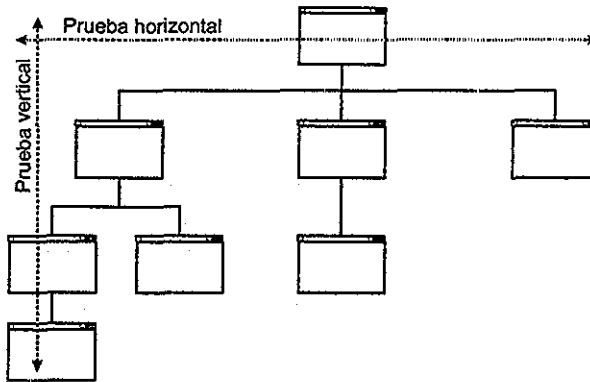


Figura 5.1: Ejemplo de un árbol de ventanas, los ejes muestran las direcciones de navegación preferenciales para una prueba vertical o una horizontal.

si hacemos que el usuario explore una parte del árbol de forma horizontal, estaremos haciendo una prueba horizontal. Esta prueba puede hacerse si queremos por ejemplo, saber si entiende las opciones más generales del sistema.

Si queremos en cambio, hacer que el usuario llegue a niveles profundos de nuestro árbol, estaremos practicando una prueba vertical. La mezcla de ambas pruebas puede hacerse si la cantidad de pantallas o ventanas navegadas no es muy grande, se puede pensar en visitar brevemente las hojas superiores y llegar a alguna hoja profunda, por ejemplo. Al prototipo que se le presenta al usuario en este tipo de prueba se le suele llamar *prototipo en T*, haciendo alusión a las formas en las cuales se puede navegar este prototipo (ver Figura 5.1).

Los desarrolladores tendrán que pensar entonces en el tiempo que durará la prueba y en el posible cansancio del usuario. Además se tendrá que pensar también en que pruebas posteriores pueden, eventualmente, alcanzar a navegar todo el árbol.

5.4.3 Pruebas exploratorias

Recién se da por terminada la, o las primeras iteraciones de la captura de requerimientos, el equipo de desarrollo piensa que se tiene claro lo que se requiere que haga el futuro sistema. Es entonces cuando se construye un prototipo, ese prototipo tiene que ser, en algún momento, evaluado.

Hablaremos entonces del primer tipo de prueba, la prueba exploratoria o de requerimientos. El objetivo de la prueba exploratoria es validar² los requerimientos del cliente. Esto es, asegurar que el prototipo del sistema hace lo que el sistema final tendrá que hacer, evaluar la efectividad de los diseños preliminares, el modelo conceptual del producto. Finalmente esto también tendrá que ver con el grado en el cual los desarrolladores captaron las ideas del cliente y el grado en el cual las entendieron. Estas pruebas entonces, darán información sumamente importante.

Esta prueba se realiza muy temprano en el proceso de desarrollo, por lo menos mientras exista la necesidad de evaluaciones a nivel de prototipo funcional, i.e, donde se modela sólo la funcionalidad del mismo.

Haciendo que el usuario interactúe con el prototipo, el equipo de desarrollo sabrá si el usuario capta y distingue los elementos de la interfaz, entiende cómo se comunican estos elementos entre sí y cómo funciona la navegación del sistema. El equipo de desarrollo también podrá tener información acerca de cómo ha manejado y plasmado en el prototipo las siguientes cuestiones:

- Organización general de la información.
- Accesibilidad del formato propuesto.
- Puntos susceptibles de ofrecer ayuda.

Las pruebas existen para contestar preguntas que los desarrolladores desean saber acerca de lo que han planeado y lo que han construido. Es el caso de este tipo de prueba, algunas de las preguntas podrían ser:

- ¿Qué piensan en general los usuarios acerca del producto?

²Validación no debe confundirse con verificación, validar se refiere a saber si los requerimientos se *plasmaron* correctamente, mientras que verificar se refiere a saber si se *entendieron* correctamente.

- ¿Entienden qué hace?
- ¿Entienden cómo funciona y cómo usarlo?
- ¿Qué partes del producto se navegan fácilmente y cuáles requieren algún tipo de ayuda?
- ¿El sistema hace lo que tiene que hacer?
- ¿Es claro y pertinente?

El equipo de desarrollo tendrá que tener entonces un prototipo con la suficiente funcionalidad como para poder responder las preguntas que se han planteado acerca de éste.

5.4.4 Pruebas de diseño

Mientras que el tipo de prueba anterior se practica a principios del desarrollo, estas pruebas se practican regularmente alrededor de la mitad del mismo (aunque esto puede variar). Su objetivo es validar las decisiones de diseño que se han tomado. Esto es, una vez que el equipo de desarrollo ha asegurado que el prototipo tiene la funcionalidad y la estructura general que se requiere, se puede pensar en evaluar las distintas soluciones que se han implementado (o que se pretende implementar). Saber cuáles de estas decisiones de diseño ayudan al usuario y cuáles no o qué tanto. Con diseño nos referimos a su significado más amplio, no sólo el diseño gráfico, sino aspectos del diseño de la interfaz que según los desarrolladores permitirá resolver problemas puntuales de la misma.

Más específicamente, con decisiones de diseño nos referimos a aspectos como ayuda en pantalla, elección, claridad y posición de los diferentes elementos, mensajes, vocabulario, colores, sonido, efectos de animación, elección de íconos, comandos, tipo de interacción, presentación de la información, etc

Todo aquello que guíe al usuario y lo ayude en el desempeño de su tarea. Las preguntas que se responderán en este caso serán como las siguientes:

- ¿Los íconos utilizados ayudarán al usuario de alguna manera?
- ¿Los sonidos mejoran la atención del usuario?

- ¿Los efectos de animación son oportunos?
- ¿Los mensajes son claros?
- ¿El usuario entiende la agrupación de los elementos?
- ¿La información y los comandos son completos y pertinentes?

5.4.5 Pruebas de desempeño

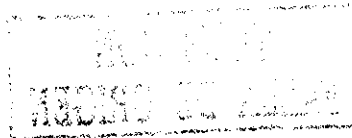
Finalmente, alrededor del final del desarrollo, se lleva a cabo una prueba de desempeño. Una vez que el equipo de desarrollo se ha asegurado que la funcionalidad del sistema es correcta y que las decisiones de diseño también lo son, se evalúa qué tan eficiente es el usuario con el sistema, i.e., qué tan bien se desempeña en situaciones de uso intensivo.

En estas pruebas, incluso puede compararse qué tan eficientemente realiza el usuario su tarea con respecto a como la realizaba antes o con un sistema anterior. Se pone al sistema en junto con el usuario en una situación de tensión, de uso real.

Supongamos por ejemplo, que el sistema tiene que responder en tiempo real ante consultas concurrentes. Previamente, el sistema ha demostrado mediante pruebas que es fácil de usar y de aprender. Supongamos ahora que se le practica una prueba de desempeño que consiste en diez usuarios, cada uno consultando el sistema de forma, intensiva. Si el sistema no responde en tiempo real ante todas estas consultas, entonces tenemos un problema de usabilidad que probablemente no se había detectado con anterioridad.

En otro ejemplo, el equipo de desarrollo pudo haber implementado ciertas animaciones que hacían atractivo e interesante el sistema. Podría ocurrir que esas animaciones resultaran molestas después de horas de trabajo continuas, o que la velocidad de trabajo disminuyera por culpa de éstas.

Esos aspectos de usabilidad del sistema que sólo pueden observarse sometiendo el producto a un uso intensivo, son los aspectos que las pruebas de desempeño están diseñadas a detectar. Habiendo presentado las pruebas con usuarios, hablaremos ahora de la infraestructura necesaria para llevarlas a cabo.



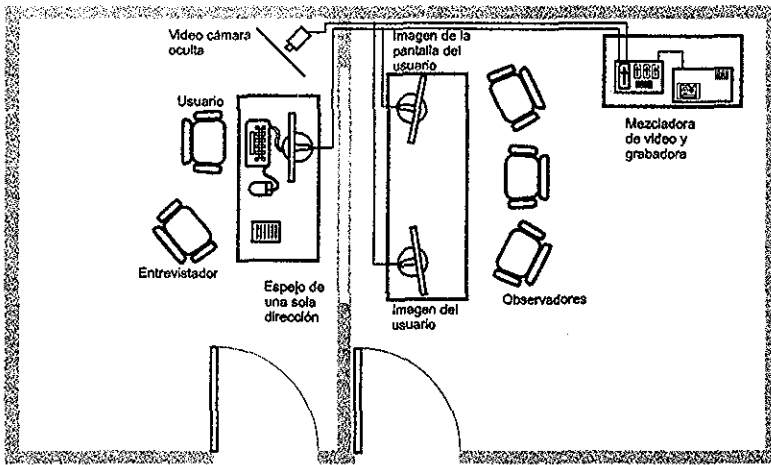


Figura 5.2: Ejemplo de laboratorio para pruebas con usuarios.

5.4.6 Preparación para la prueba

¿Qué necesita el equipo de desarrollo de software para realizar pruebas con usuarios? Primero que nada necesita un laboratorio para tales propósitos. Lo ideal es que se cuente con un lugar ex profeso, si el equipo no cuenta con este lugar, cualquier cuarto podría ser acondicionado para pruebas de suficiente calidad. Además del laboratorio, también se necesitarán los recursos humanos necesarios para implementarlas.

El laboratorio

Existen una infinidad de configuraciones posibles para este tipo de laboratorios [Rubin, 1994], escogeremos una que a nuestro juicio es la que más ventajas ofrece, tanto en términos de recursos como en términos de efectividad. Llamaremos a esta configuración de *Observación electrónica simple*, misma que se muestra en la Figura 5.2.

Además de las instalaciones, es necesario contemplar a los participantes

TESIS CON
FALLA DE ORIGEN

involucrados: los observadores (a quienes les interesa observar la prueba), el usuario (quién interactuará con el prototipo) y el entrevistador (quien dará instrucciones y guía al usuario). Más adelante hablaremos con más detenimiento de ellos.

En esta configuración de laboratorio, los observadores se encuentran ocultos en un cuarto anexo, pueden observar, pero no pueden comunicarse directamente con el entrevistador. El usuario no tiene porqué saber de antemano que éstos existen ni que sus acciones y su persona serán grabadas³. El entrevistador guiará al usuario a través de la prueba, proporcionándole todo lo que éste necesite y que no sesgue el resultado de la prueba.

Los observadores podrán registrar todo evento que ocurra en la sala de pruebas. Será necesario, por lo menos, registrar la pantalla del prototipo, el audio ambiental y del prototipo, así como al usuario mismo. Podría ser útil también en determinados casos, registrar también las acciones mismas del usuario como teclado y movimientos del mouse.

Si se tienen los recursos para montar correctamente este tipo de laboratorio, la única desventaja importante de esta configuración de laboratorio sería que el comportamiento del entrevistador podría afectar en algunos casos el resultado de la prueba. El entrevistador podría dar ayuda extra al usuario o incomodar a éste con algunos comentarios y/o actitudes negativas.

En otro tipo de configuraciones, los observadores están presentes junto con el usuario y el entrevistador. Esto no es deseable debido a que suele inducir nerviosismo en el sujeto. En el otro extremo de configuración, el entrevistador se encuentra junto con los observadores en un cuarto aparte y se comunica con el sujeto mediante un micrófono y altavoces. Esto tampoco es deseable ya que la voz del entrevistador puede imponer al sujeto, promoviendo su nerviosismo y sesgando, por consiguiente, la prueba.

³En los lugares geográficos que por alguna razón (por ejemplo legal), el sujeto tenga que saber de antemano estas cuestiones, se tendrán que tomar las medidas y los cambios a la prueba correspondientes.

Los trabajadores

Ingeniero de pruebas de usabilidad es el encargado de planearla, diseñarla, escoger al entrevistador y es, en general, la autoridad responsable durante el desarrollo de la misma.

El monitor es aquel que tiene contacto directo con el usuario, le da la bienvenida, trata de hacerlo sentir tranquilo y le guía en la prueba. La elección de este trabajador es especialmente importante, el éxito o fracaso de una prueba con usuarios depende en gran medida del desempeño del entrevistador. Este no podrá ayudar al usuario más allá de lo previamente acordado y evidentemente no podrá burlarse, mostrar desesperación, frustración y cualquier otra actitud negativa.

El monitor recibirá ayuda extra (en caso de una eventualidad) por parte de los observadores o del administrador sólo a través de un dispositivo de comunicación a distancia. Esta comunicación tiene que ser lo más discreta posible. Adicionalmente, el entrevistador no tiene por qué tener ningún conocimiento del sistema, ni tiene por qué ser del área de cómputo. De hecho, es preferible que tenga experiencia en el trato con personas, que tenga buena dicción, sea amable, etc.

Los observadores por su parte, conocerán muy bien el sistema y en general serán parte del equipo de desarrollo. La elección de estos observadores es también importante ya que previo a la prueba colaborarán con el administrador para plantearla y diseñarla. Durante la prueba misma colaborarán con el entrevistador (a distancia), para prestarle ayuda en caso de eventualidades. Estos observadores tendrán que tener un conocimiento completo en aquellas partes claves del sistema que desean probar.

5.4.7 Las seis etapas de la prueba

Haremos ahora una breve revisión de las etapas en las que se dividen las pruebas de usabilidad con usuarios:

- Elaboración del plan de pruebas.
- Elección de los participantes.
- Creación de los artefactos.

- Realización de la prueba.
- Despedir al usuario.
- Analizar la prueba.

Elaboración del plan de pruebas

Cuando se decide que una prueba con usuarios es requerida, es porque se tienen ciertas hipótesis acerca del prototipo actual que se quieren probar. Con estas hipótesis se elabora un plan de pruebas. Para un ejemplo de un plan de pruebas referirse al Apéndice A.

El plan de pruebas es un documento conciso que sirve como preludeo a la prueba. El formato de este plan queda a discreción de la parte del equipo de desarrollo que se encarga de las pruebas. A continuación mencionaremos algunas secciones importantes que debería de tener:

- *Motivaciones.* Aquí se debe explicar (coloquialmente), y de manera breve qué motivó la prueba.
- *Hipótesis u objetivos.* Se mencionarán todas las hipótesis que se tienen acerca del sistema y que se quieren verificar. Ejemplo: *El usuario entenderá y sabrá utilizar la barra de navegación.*
- *Perfil del usuario.* Se describirán las características que tendrán (o que deberían tener) los usuarios que se requerirán para la prueba.
- *Método.* Contendrá una descripción de cómo se piensa llevar a cabo la prueba, i.e., la metodología.
- *Lista de tareas.* Las tareas que se le solicitarán al usuario que realice. Estas tareas tienen que estar dirigidas a verificar la validez de las hipótesis planteadas anteriormente. Tomando el ejemplo anterior, una tarea puede ser: *Por favor navegue por el sistema utilizando la barra de navegación.*
- *Protocolo de bienvenida.* Este documento es muy importante, es básicamente el texto que el entrevistador leerá por igual a todos los participantes durante la prueba. Contendrá su propia presentación, aclaraciones pertinentes, instrucciones generales y las tareas a realizar. Es importante

que el texto se lea para asegurar que todos los participantes reciban exactamente la misma información.

- *Condiciones de la prueba*. Aquí se describirán los materiales necesarios, el ambiente en el que se desarrollará y el equipo requerido.
- *Especificación del instrumento*. Se explica qué cantidades serán medidas en la prueba: Tiempo, satisfacción del usuario, tasa de errores, etc.
- *Formatos de resultados*. Estos pueden generarse al término de la prueba pero es una buena idea ponerlos aquí ya que ayudarán a estandarizar futuras pruebas.

Elección de los participantes

La selección de los participantes es crucial para el proceso de prueba: Si se prueba a las personas erróneas, la prueba no habrá tenido ningún sentido.

Primero que nada se tienen que tener caracterizados a los usuarios (edad, sexo, lengua, formación, experiencia con computadoras y con el trabajo que realiza, etc.). Para algunos sistemas, incluso la raza puede resultar vital. Si es posible, estos usuarios deben ser los futuros usuarios del producto final, i.e, si el sistema está diseñado para que lo usen niños de ocho años, los usuarios tienen que ser niños de ocho años.

Teniendo caracterizados a los usuarios, se debe conocer de dónde pueden ser elegidos y realizar, para ese grupo, una selección aleatoria para la prueba. A estos usuarios que ya escogimos, los tenemos que conocer muy bien. Toda información referente a ellos puede ser útil. Si existe algún mandamiento respecto al diseño de interfaces este es:

Conoce a tu usuario

Creación de los artefactos

¿Qué artefactos⁴ necesitamos para la prueba? Tomemos en cuenta que los artefactos no son lo mismo que los materiales que se usarán en la prueba. Hablemos brevemente de los materiales:

Supongamos que queremos que nuestro usuario ingrese varios registros mediante nuestra interfaz a una base de datos. No sería correcto que el entrevistador se los dictara. Recordemos que estamos tratando de simular las condiciones de operación de dicho sistema; evidentemente en el ambiente real no habrá un entrevistador junto a los operadores

La solución es dar al usuario información impresa en el formato habitual con el cual el usuario trabaja. Este material impreso forma parte del tipo de material que se podría necesitar para la prueba. Podría ser que el usuario necesite papel para imprimir, discos vírgenes para grabar, etc.

Ahora, los artefactos son piezas de información utilizada por los trabajadores, no por el usuario. Más adelante hablaremos de los artefactos de las pruebas con bastante profundidad, ahora, sin embargo mencionaremos los que son utilizados en la prueba misma:

- *Cuestionario de perfil del usuario.* Tendrá las preguntas que darán a los desarrolladores información sobre el tipo de usuarios que sistieron a la prueba.
- *Protocolo de bienvenida.* El documento que será leído por la persona que estará con el usuario durante la prueba.
- *Plan de pruebas de usabilidad.* Este documento contendrá principalmente las actividades a ser realizadas por el usuario durante la prueba.
- *Cuestionario de usabilidad.* En el caso de que la prueba esté encaminada a determinar el desempeño de la interfaz, éste cuestionario proveerá una manera de medir este desempeño como es percibido por el usuario.
- *Prototipo de la interfaz.* El prototipo que usará el usuario

⁴Por consistencia en el lenguaje del Proceso Unificado, introduciremos aquí el término artefacto: *Toda pieza de información que es utilizada o producida por un proceso de desarrollo* [Jacobson et al, 1999].

Realización de la prueba

En una prueba clásica ocurren los siguientes eventos:

1. Previamente el usuario pudo haber contestado ya el cuestionario de perfil del usuario
2. El monitor recibe al usuario, le explica brevemente en lo que consistirá la prueba y lo instala en el laboratorio.
3. Cuando el usuario está listo y se sienta preparado, el entrevistador lee el protocolo de bienvenida y le indica al usuario las tareas a realizar

Despedir al usuario

Al terminar la prueba, el entrevistador da las gracias al usuario y, si no lo ha hecho antes, le proporciona el cuestionario de perfil del usuario para que lo conteste. Adicionalmente, el entrevistador puede aclarar algunos aspectos de la prueba que no se podían aclarar durante ésta o contestar algunas preguntas hechas por el usuario durante la prueba y que no se podían contestar en ese momento.

Analizar la prueba

Ya que se han llevado a cabo las pruebas con todos los usuarios, llega el momento de analizar toda la información recabada en el proceso. El material queda registrado comúnmente en videos grabados durante la prueba y muestran al usuario desempeñándose con el prototipo y su pantalla, así como el audio de la prueba.

Los videos son analizados a profundidad por los observadores (preferentemente) y la persona específicamente designada para realizar este análisis. Extraerán toda la información importante: Las fallas funcionales del prototipo, los comentarios de los usuarios y sus propios comentarios (puede ocurrir que durante el análisis del material, sea notorio un error que no fue mostrado por el sistema ni mencionado por el usuario).

Después de que cada observador haya analizado el video y extraído la información importante, se tiene que crear el reporte de prueba de usuario. En éste, se registrará el tiempo que duró la prueba, los tiempos que tardaron las

tareas y su grado de compleción, los errores del prototipo, los comentarios y errores del usuario y los comentarios de los observadores tanto durante, como después de la prueba.

Este documento es vital ya que será el punto de partida para una nueva iteración de desarrollo, i.e, los requerimientos generados por las pruebas serán los nuevos requerimientos de la nueva iteración.

Hasta aquí hablamos de las pruebas de usuario en el contexto del desarrollo de software, las metodologías, el equipo, los trabajadores y los productos. Ya que uno de los dos temas principales de esta tesis son las pruebas con usuarios, nuestro propósito aquí era proporcionar suficiente información para entender cabalmente este tipo de pruebas.

5
6
7

Parte III

**Pruebas de usuario y Proceso
Unificado**

62

Capítulo 6

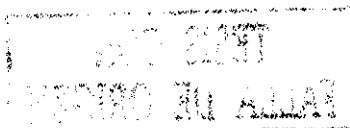
El Proceso Unificado y la interfaz

En el Capítulo 4 hicimos una breve revisión de la usabilidad en el PU. En este trabajo se propondrá una forma de añadir al PU las pruebas con usuarios de manera que el proceso siga siendo esencialmente el mismo ante los ojos del desarrollador.

Para lograr este objetivo primero tenemos que entender la filosofía actual con la que el PU maneja la usabilidad y las pruebas de usuario. Esta filosofía se tendrá que respetar lo más posible en el capítulo siguiente

6.1 La interfaz de usuario en el Proceso Unificado

Debido a su importancia, volvemos a mostrar la figura que resume al PU (Figura 6.1): Toda interfaz debe empezar con la fabricación de un prototipo, en [Jacobson *et al*, 2000] se especifica que el prototipo debe construirse en la fase de construcción (lo cual, aunque suene obvio no lo es). Sólo se “permite” realizar este prototipo en alguna fase anterior si se necesitaba un nuevo tipo de interfaz o realizar una demostración. Ahora, hablando en términos de flujos de trabajo, el diseño de la IGU se debe realizar durante la captura de requisitos (aunque el nombre de la actividad sugiera otra cosa). Vista como un “accesorio” de los casos de uso, los analistas deben especificar cuál será el aspecto de la IGU. Se sugiere realizar esbozos de la interfaz, éstos



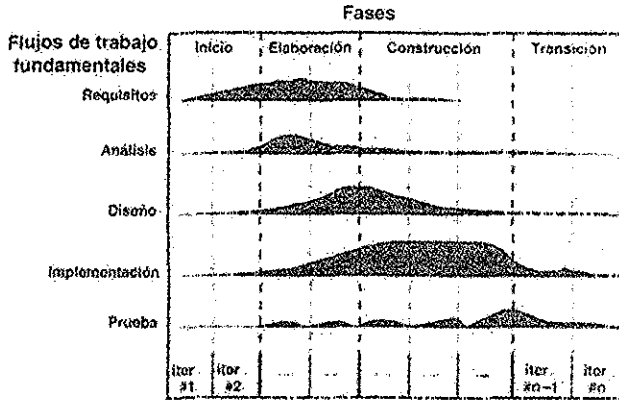


Figura 6 1: Representación gráfica del Proceso Unificado

esbozos deberán mostrar la información que será manejada. Es bueno hacer varios esbozos, discutirlos con los usuarios y construir prototipos para que los usuarios los prueben. Este *prototipo de interfaz de usuario* (PIU) constituye un artefacto bien definido dentro del PU. Así mismo, el trabajador que construye este artefacto también se encuentra definido en el proceso y lleva por nombre *Diseñador de interfaz de usuario* (DIU).

6.2 Diseño de la interfaz de usuario

Habiendo definido qué se construirá y quién lo hará hay que saber cómo hacerlo. El PU especifica el procedimiento para ir de los casos de uso al prototipo de la interfaz. Esta acción es la que realiza el DIU para crear el PIU.

6.2.1 De las reglas del negocio a la interfaz

Ya hemos dicho que el diseño de la IGU debe realizarse durante la captura de requisitos según el PU. Bien, supongamos entonces que tenemos ya los diagramas de casos de uso. De lo que se trata es de "mapear" los casos

TESIS CON
 FALLA DE ORIGEN

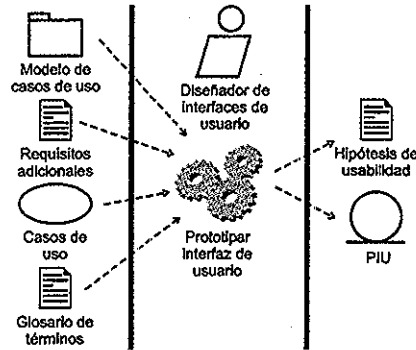


Figura 6.2: Actividad: Prototipar interfaz de usuario. Los estereotipos mostrados corresponden a los utilizados en [Jacobson *et al*, 2000].

de uso en elementos de interfaz¹. En este punto se sugiere que al hacer el levantamiento de casos de uso se evite ser demasiado explícitos en cuanto a funcionalidad se refiere. Por ejemplo, no es bueno describir en los casos de uso qué información se capturará antes que otra en la misma forma, tampoco es bueno hablar *a priori* de los diferentes elementos de la interfaz. Esto no es fácil ya que el equipo de desarrollo siempre procura ser sumamente específico en las descripciones de casos de uso y tiende a mencionarla en los mismos.

Se hace entonces un diseño lógico de la interfaz, discerniendo lo que se necesita de las interfaces para posibilitar la realización de los casos de uso para cada actor. Esta actividad se realiza tomando información del modelo de casos de uso, el glosario de términos, los casos de uso (descritos) y los requisitos adicionales. Esto se muestra en la Figura 6.2.

Ejemplo:

1. En nuestro glosario de términos aparecen los términos *Nombre del cliente* y *Ventana de captura del Nombre del cliente*.
2. En nuestro diagrama de casos de uso se expresa: “El cliente (actor)

¹Nótese que un mapeo es capaz de producir la funcionalidad pero de ningún modo puede garantizar la usabilidad o la reactividad requerida de la interfaz.

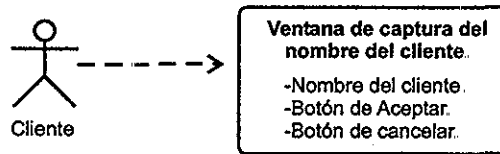


Figura 6.3: Diseño lógico de la interfaz del ejemplo. Nótese que se registra el nombre del elemento principal así como sus "atributos", los cuales son así mismo elementos de interfaz

proporciona su nombre para ser buscado".

3. Como requisitos adicionales tenemos: "El cliente sólo debe de preocuparse de proporcionar su nombre"

Con base en esta información, el DIU puede traducir "Nombre del cliente" en un campo de texto apoyándose en los puntos 1 y 2. En la Figura 6.3 se muestra un posible diseño lógico para el ejemplo planteado. Como apoyo a su actividad, el DIU puede, a sugerencia del mismo PU, hacerse las siguientes preguntas para cada actor:

- ¿Qué elementos de interfaz de usuario se necesitan para posibilitar el caso de uso?
- ¿Cómo deberían relacionarse unos con otros?
- ¿Cómo se utilizarán en los diferentes casos de uso?
- ¿Cuál debería ser su apariencia?
- ¿Cómo deberían manipularse?

Así mismo, para determinar qué elementos de interfaz de usuario necesitan ser accesibles al actor en cada caso de uso, el DIU puede contestar las siguientes:

- ¿Qué clases del dominio, entidades del negocio o unidades de trabajo son adecuadas como elementos de la interfaz de usuario para cada uso?
- ¿Con qué elementos de la interfaz de usuario va a trabajar el actor?

- ¿Qué acciones puede invocar el actor, y qué decisiones puede tomar?
- ¿Qué guía o información va a necesitar el actor antes de invocar cada acción de los casos de uso?
- ¿Qué información debe proporcionar el actor al sistema?
- ¿Qué información debe proporcionar el sistema al actor?

Cabe mencionar que la elección y disposición de los elementos de interfaz queda, por lo menos en una primera instancia, a criterio del DIU. La actividad continuará hasta tener un conjunto de esquemas de interfaces de usuario.

6.2.2 El diseño físico

Habiendo creado el diseño lógico de la interfaz, el siguiente paso es evidentemente crear un diseño físico. Un diseño físico puede ser desde bosquejos en papel hasta prototipos funcionales. Lo primero que se hace generalmente es bosquejar la futura interfaz con base en el diseño lógico. Primero se pueden bosquejar los elementos necesarios y después se pueden generar bosquejos completos, añadiendo elementos adicionales generalmente requeridos.

Una vez que se acuerda la interfaz a ser construída, se puede generar un prototipo ejecutable generalmente haciendo uso de herramientas de prototipado rápido o RAD. Se sugiere hacer varios prototipos, uno para cada actor, para que cada uno ejecute el caso de uso que le corresponde. Notaremos especialmente que el PU sugiere crear ejecutables sólo para los actores más importantes, i.e., cuando se tiene mucho que ganar en usabilidad. La creación del diseño físico es encontrar la mejor manera de mapear el diseño lógico en un diseño físico. En el caso de nuestro ejemplo, el pasar del diseño lógico mostrado en la Figura 6.3 al diseño físico podría fácilmente darnos como resultado la interfaz que mostramos en la Figura 6.4.

Una vez generados los prototipos (ejecutables o no) pasan a ser revisados por los *revisores*, los cuales deben verificar que la interfaz

- Permita que el actor navegue de forma adecuada

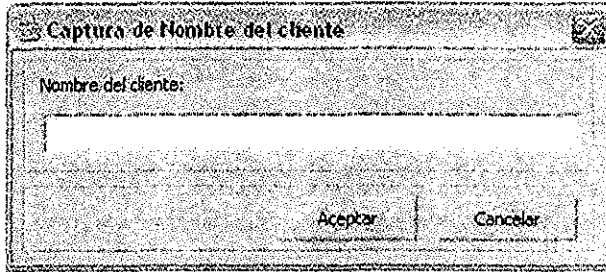


Figura 6.4: Posible diseño físico resultante del diseño lógico de la Figura 6.3.

- Proporciona una apariencia agradable y una forma consistente de trabajo con la interfaz de usuario, como por ejemplo, en cuanto a orden de tabulación y teclas rápidas
- Cumpla con los estándares relevantes como el color, tamaño de los botones y situación de las barras de herramientas.

6.2.3 Las pruebas

Habiéndose expuesto la mecánica a seguir para pasar de las reglas del negocio (los casos de uso) al PIU, el PU abandona el tema aunque no definitivamente. Así, el equipo de desarrollo debe entender cómo manejar la construcción de un prototipo de IGU durante la captura de requerimientos.

Si volvemos a la Figura 6.1 observaremos que el PU posee dos “dimensiones” para localizar un punto específico en el desarrollo del proyecto. La dimensión de los flujos de trabajo y la de las fases. Bien, los desarrolladores pueden confundirse y pensar equivocadamente que la realización del prototipo debe ocurrir al inicio del proyecto tan pronto como se tengan los casos de uso. Para el PU esto está generalmente mal, ya que se debe tener en cuenta de que el PU es “bidimensional”.

Como ya se explicó en la Sección 6.1, el trabajo de diseñar y construir el PIU debe ocurrir hasta la fase de construcción. Lo que ocurre en esta fase se aborda ya tarde en el PU y en efecto se le dedica un poco más atención a

la IGU

En efecto, en fases posteriores se explica que se debe probar el (o los) prototipos de IGU creados, éstas pruebas deben ser con usuarios reales, mismos que probarán estos prototipos. Basándose en sus respuestas, el equipo de desarrollo deberá amoldar el prototipo hasta satisfacer las necesidades de los usuarios.

Hay que mencionar también que el PU no considera indispensable ni para todos los casos la construcción de un prototipo de interfaz: *Para sistemas de los que esperemos un gran número de ventas, construiremos un prototipo de interfaz de usuario incluso si no es muy compleja. El coste de reemplazar una interfaz poco satisfactoria una vez distribuido el producto sería muy grande* [Jacobson et al, 2000].

Nuestra propuesta difiere de ésta idea, nosotros planteamos que en *todo caso* se debe realizar un prototipo de interfaz. Ya que en fases posteriores puede ser demasiado tarde, incluso pueden cambiar los requisitos. Esto debe hacerse desde el inicio y a lo largo de todo el proyecto e incluso si la interfaz es sencilla, ya que es la única forma sistemática en la cual podemos asegurar la usabilidad de nuestros productos de software.

Capítulo 7

Pruebas y Proceso Unificado

Este es sin duda el capítulo más importante de este trabajo. En el mismo, desarrollaremos nuestra propuesta: El como inyectar en el PU las pruebas de usabilidad requeridas, de manera que el equipo de desarrollo se asegure de estar contruyendo software realmente usable.

7.1 Las pruebas a lo largo de todo el proceso

Los que se dedican a IHC siempre han insistido en que la usabilidad es algo que se tiene que cuidar durante todo el proceso de desarrollo. Ya que la interfaz tiene especial impacto en la usabilidad de un producto de software, el equipo de desarrollo debe cuidar la usabilidad de la misma lo más temprano que se pueda en el desarrollo.

Nosotros proponemos una forma de elevar la usabilidad de la IGU mediante pruebas sistemáticas con usuarios En el contexto del PU. Es entonces necesario desde el punto de vista de IHC, que se trate de llevar estas pruebas a cabo desde la fase de inicio del proyecto, hasta la fase de transición.

El PU especifica, como ya dijimos, que la construcción de un primer prototipo de IGU debe suceder en la fase de construcción Sin embargo, nosotros proponemos diseñar y si es posible implementar este prototipo desde la fase de inicio.

Como en la fase de inicio ocurren generalmente dos iteraciones a lo más,

habrá ocasiones en las cuales sea imposible tener algún diseño de la interfaz tan temprano. Los primeros diseños deben ocurrir entonces en la fase de elaboración cuando más tarde. Es en esta fase cuando se aplicarán las pruebas exploratorias como ya hemos mencionado; las pruebas de diseño y desempeño se llevarán a cabo en las fases de construcción y transición respectivamente.

Esencialmente, el equipo de desarrollo y especialmente el DIU tienen que tener siempre en mente que la IGU debe tener una presencia en el desarrollo, desde el inicio del mismo hasta su terminación. Lo mismo debe ocurrir entonces con las pruebas de usabilidad si es que se quiere realmente lograr un sistema fácil de usar y con el cual se pueda desempeñar eficientemente el usuario

7.2 Artefactos, Trabajadores y Actividades

En el capítulo anterior vimos cómo ataca el PU la construcción del prototipo de la IGU. Como ya hemos explicado anteriormente, el equipo de desarrollo debe tener una manera de saber qué tan bien ha diseñado esta interfaz.

El contexto en el cual plantearemos las adiciones al PU es el siguiente: Un equipo de desarrollo (grande o chico) participa en un proyecto de software cuya metodología está basada en el PU y desea aumentar la usabilidad del sistema que va a construir. Primero que nada, requeriremos que se tenga ya un primer prototipo de la IGU del futuro sistema, más adelante veremos en qué momento debemos construir esta interfaz, misma que deberá probarse en el flujo de trabajo "prueba" inmediato a la aparición del prototipo.

No todas las pruebas serán iguales a lo largo de todo el desarrollo. Como ya explicamos en el Capítulo 5, existen tres tipos de pruebas que serán de nuestro interés. Por ahora sin embargo, nos olvidaremos de estos diferentes tipos de prueba y pensaremos simplemente en lo que se necesita para llevarlas a cabo en el último de los flujos de trabajo: El de prueba.

7.3 Artefactos

Como sabemos, en el PU, hay actividades que son llevadas a cabo por trabajadores. Estos trabajadores utilizan un conjunto de artefactos para realizar su actividad y generan otro. A continuación introduciremos los artefactos necesarios para llevar a cabo las actividades que permiten asegurar la usabilidad de un sistema en el contexto del PU.

7.3.1 Artefacto: Hipótesis de usabilidad

El PU especifica que el *Diseñador de interfaces de usuario* (DIU) debe diseñar la IGU tomando como base los casos de uso, el modelo de casos de uso, requisitos adicionales y el glosario de términos (ver Figura 6.2). Este trabajador tiene que tener ciertas metas que quiere lograr con esta interfaz (la cual desde su punto de vista es una buena interfaz). Algunas metas pudieran ser las siguientes:

- Lograr una navegación sencilla y fácil de usar.
- Permitir que el usuario desempeñe su tarea eficientemente
- Lograr un aspecto atractivo
- Hacer intuitivo el entendimiento de la interfaz.

El diseñador entonces realizará su actividad (Construir prototipo de la interfaz de usuario) con estos objetivos en mente. Al terminar, el DIU tendrá ciertas hipótesis acerca de la IGU que acaba de construir (véase también Sección 5.4.3). Para las metas anteriores, las hipótesis podrían ser las siguientes:

- La interfaz será sencilla y fácil de usar
- El usuario desempeñará su tarea más eficientemente que con el sistema anterior.
- Al usuario le parecerá atractiva la interfaz.
- El usuario recurrirá rara vez al uso de ayudas contextuales.

El DIU deberá plasmar estas hipótesis en un documento detallado que será utilizado por el *Ingeniero de pruebas* (IP).

7.3.2 Artefacto: Plan de pruebas de usabilidad

El PU especifica que cuando el ingeniero de pruebas planifica una prueba del sistema, se produce un plan de pruebas. Nosotros haremos una distinción entre las pruebas del sistema y las pruebas de usabilidad. Si bien en esencia se prueban en ambos aspectos del sistema, la metodología para llevar a cabo ambos tipos de prueba son diametralmente diferentes. La razón es que mientras que en las pruebas del sistema los desarrolladores lidian con un producto de su creación, en las pruebas de usabilidad se lidia con humanos.

El ingeniero de pruebas deberá entonces ser responsable de dos planes de pruebas por cada iteración. El plan de pruebas y el *Plan de pruebas de usabilidad*. Este nuevo artefacto deberá contener toda la información necesaria para llevar a cabo la prueba de usabilidad. En concreto deberá contener:

1. Breve descripción del sistema.
2. Las justificaciones (o motivaciones) para llevar a cabo la prueba de usabilidad.
3. Hipótesis de usabilidad modificadas a discreción del ingeniero de pruebas: Probablemente, el ingeniero de pruebas juzgue pertinente hacerlas más concisas o cambiar el vocabulario con vista al entendimiento por parte del trabajador que llevará a cabo tal prueba.
4. Requerimientos especiales para la prueba. Por ejemplo: Computadora conectada a Internet, monitor de 19", etc. Ver Apéndice A, sección A.1.

7.3.3 Artefacto: Cuestionario de perfil del usuario

Dos aspectos importantes dentro de la planeación de una prueba de usabilidad con usuarios reales son los siguientes:

- Garantizar que los usuarios que se presenten a la prueba tengan el perfil requerido.
- Tener información adicional de los usuarios que nos permita entender mejor sus acciones durante la prueba.

Una buena manera de obtener esta información es por medio de un cuestionario que los usuarios tendrán que llenar al inicio o al final de la prueba. En este cuestionario los usuarios responderán preguntas encaminadas a conocer su experiencia en el área de aplicación del sistema, así como su experiencia con computadoras.

Mientras que tener información acerca de su experiencia en el negocio del sistema (por ejemplo contabilidad, en el caso de ser un sistema contable) es evidentemente importante, los desarrolladores también necesitarán saber la experiencia general del usuario con computadoras. Ciertamente será útil conocer si se manejan bien con el mouse, si entienden lo que es un campo de texto, si entienden la mecánica de las tabulaciones en un formulario, etc.

Errores que surgieran durante la prueba, podrían ser ocasionados por problemas relacionados con el manejo de computadoras por parte del usuario y no por su conocimiento acerca del negocio. Entonces, el cuestionario de perfil del usuario es un documento con una serie de preguntas encaminadas a obtener este tipo de información. El IP será el encargado de confeccionar este cuestionario. Un ejemplo de este artefacto se puede revisar en el anexo. Ver apéndice A, sección A.2.

7.3.4 Artefacto: Protocolo de bienvenida

En el Capítulo 5, explicamos la naturaleza y consistencia de este artefacto (Sección 5.4.3). Como sabemos, será leído por el monitor de pruebas de usabilidad y contendrá información sobre la prueba, misma que es importante al usuario. Adicionalmente, el protocolo de bienvenida tiene como objetivo tranquilizar al usuario, hacerlo sentir relajado para que la prueba simule realmente las condiciones de trabajo que se esperan con el sistema final. Ver Apéndice A, sección A.3.

7.3.5 Artefacto: Tareas de usuario

Este documento deberá contener, de la forma más clara posible, las actividades que el usuario llevará a cabo durante la prueba de usabilidad.

Estas actividades o *tareas* deberán estar lo suficientemente detalladas como para poder ser entendidas tanto por el usuario que participará en la

prueba, como por el trabajador que lo acompañará durante la misma. Adicionalmente, estas tareas deberán estar acompañadas de las fichas o tarjetas (con la información requerida) que le serán dadas a los usuarios para que puedan llevar a cabo sus actividades con el prototipo.

Por ejemplo, supongamos que el usuario requerirá dar de alta un par de registros a una base de datos para probar la facilidad de uso del prototipo. Evidentemente necesitará que las fichas tengan la información que será dada de alta en la base de datos. En algunos casos es recomendable que en vez de darle fichas, se le de la información en el formato en el cual habitualmente lo tienen en su trabajo diario (hojas con un formato específico, cheques, facturas, etc).

Las actividades deberán ser compatibles con las hipótesis de usabilidad que aparecen en el *plan de pruebas de usabilidad*. Ver Apéndice A, sección A.4.

7.3.6 Artefacto: Instrumento

Cuando se realiza una prueba se obtienen valores, éstos valores deberán surgir de aquello que se le pida al usuario y de cómo éste reaccione ante las respuestas del sistema. En algunos casos nos interesará saber el tiempo que le tomó al usuario llevar a cabo una tarea, en otras nos interesará saber si el usuario logró o no ciertos objetivos (obtendremos valores SI y valores NO), y en otras nos podría interesar el número de elementos que logró entender de cierta pantalla. El documento que especifica qué es lo que vamos a medir y cómo lo vamos a medir es el *Instrumento*.

En ciertos casos nos podría interesar medir el grado en el cual una interfaz resulta atractiva para el usuario. Como esta es una medición subjetiva, el instrumento podría ser alguna escala empírica que diera información sobre el grado de gusto por parte del usuario [Bevan, 1999]. Por ejemplo, el grado de satisfacción puede medirse mediante cuestionarios que han sido probados en sistemas reales, un ejemplo es el cuestionario SUS [Kirakowski]. Ver Apéndice A, sección A.5.

7.3.7 Artefacto: Cuestionario de usabilidad

El cuestionario de usabilidad es un cuestionario que preferentemente se debe de aplicar en toda prueba de usabilidad. Este cuestionario está diseñado para proporcionar una manera de calificar la calidad de la IGU en términos de usabilidad.

Este cuestionario es contestado por el usuario justo después de la prueba, y en él asienta su opinión subjetiva acerca de la usabilidad del sistema. Entre más grande sea el número de cuestionarios contestados para una misma serie de pruebas, más significado tendrán los resultados estadísticos que se desprendan de las mismas.

Existen varios tipos de cuestionarios, algunos gratis y otros no, que proveen al equipo de desarrollo de una manera de saber si el sistema se desempeña adecuadamente. Tres ejemplos de cuestionarios que han sido utilizados exitosamente en proyectos reales son MUSIC [SUMI], SUS [Kirakowski] y QUIS [Harper, 1993]. En este trabajo utilizaremos el cuestionario SUS por ser sencillo y gratuito, además de haber sido probado con éxito en proyectos de desarrollo desde hace más de 10 años. Ver Apéndice A, sección A.6.

7.3.8 Artefacto: Registro de prueba de usabilidad

Recordemos que un artefacto es prácticamente todo aquello que se produce durante el desarrollo y que los trabajadores utilizan para realizar sus actividades. En general, los artefactos son documentos, particularmente, diagramas, textos, dibujos, gráficas, etc.

El *Registro de prueba de usabilidad* es en general una videograbación de la prueba de usabilidad misma. Como explicamos en el Capítulo 5, la prueba, además de ser observada en tiempo real, también se graba para su posterior análisis. Lo que interesa al equipo de desarrollo es el desempeño del usuario con el prototipo, por lo tanto, sólo será interesante tener grabado al usuario, la pantalla que el mismo ve y el audio completo de la prueba.

Podría ser, sin embargo, que se requiriera de registros adicionales. Supongamos que es vital conocer exactamente la tasa de captura del usuario o bien, las teclas precisas que presiona durante la prueba. En este caso tendría que haber un programa secundario diseñado para capturar esta información

en tiempo real y escribirla en un archivo de texto para su análisis.

En un caso como el anteriormente descrito, además de la, o las videograbaciones con audio, se tendría un archivo de texto con la información capturada automáticamente durante la prueba. El artefacto entonces, es el conjunto de todos los registros generados en la prueba

7.3.9 Artefacto: Observaciones de prueba de usabilidad

Durante la prueba de usabilidad, existen observadores cuyo propósito es verificar que la prueba se llevó a cabo como debía de ser, así como presenciar ellos mismos el desempeño de su interfaz.

Una vez terminada la prueba, los observadores tendrán que generar un documento en el cual plasmen toda observación que consideren importante para el análisis. Podría ser que determinaran que tal pieza de hardware dificultaba el desempeño del usuario o que las condiciones bajo las cuales se llevó a cabo la prueba no simulaban adecuadamente las condiciones para las cuales fue diseñado el prototipo.

Cabe mencionar que estas observaciones son sólo observaciones, i.e, si los observadores llegaron a considerar la prueba un fracaso, sólo podrían expresarlo, pero sería el IP el que tendría la decisión final. Evidentemente, las observaciones de la prueba de usuario deberán de tener influencia en el ingeniero de pruebas, finalmente fueron los diseñadores de la interfaz de usuario los que la crearon.

7.3.10 Artefacto: Resultados de prueba de usabilidad

Este artefacto es un paquete, i.e, contiene más artefactos (Figura 7.1). En sí, los *Resultados de prueba de usabilidad* es el artefacto que finalmente entrará a la siguiente iteración con las recomendaciones para el diseño de la interfaz. Este artefacto es producido por el *Analista de pruebas de usabilidad* (APU). Existen formatos estandarizados para reportar usabilidad, el equipo, si lo desea, puede hacer uso de estos formatos, un ejemplo es el *Industry Usability Reporting* [IUSR].

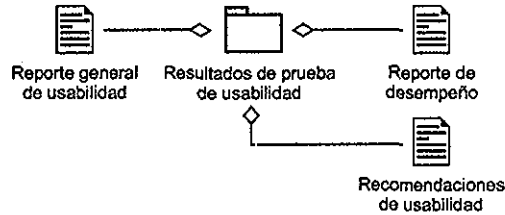


Figura 7.1: Entrada y resultado y la planeación de la prueba de usabilidad.

Artefacto: Reporte de desempeño

El *Reporte de desempeño* es un documento que contiene las observaciones del APU con respecto al desempeño que el usuario tipo tuvo con el sistema.

Supongamos que a pesar de ser fácil de usar, la mecánica de uso del programa hace que el trabajo de los usuarios se realice por arriba del tiempo planteado en la especificación de los instrumentos. O supongamos que el sistema no responde tan rápido como lo suponen las hipótesis de usabilidad. Bien, todo esto se documentará en el reporte de desempeño. Además, el reporte de desempeño podrá evidenciar aspectos ajenos al sistema que finalmente tienen su impacto en el desempeño del mismo, por ejemplo: El ancho de banda en un sistema distribuido.

Observemos que el reporte de desempeño existirá generalmente en el caso de que se haya realizado una prueba de desempeño como la explicamos en el Capítulo 5. Como en el caso del cuestionario de usabilidad, sin embargo, es posible que el reporte de desempeño exista en pruebas anteriores si los trabajadores involucrados así lo requieran.

Artefacto: Recomendaciones de usabilidad

Este artefacto es un documento importante, ya que será este documento el que contendrá lo que habrá que cambiar en las implementaciones posteriores de la interfaz.

Se deberá explicar detalladamente qué cambios se sugieren al diseño de la interfaz y las razones. Los diseñadores de la interfaz deberán hacer todo lo

posible por llevar a cabo los cambios establecidos en el documento, sin embargo, podría haber situaciones en las cuales no fuera posible implementar los cambios en el nuevo prototipo.

Podría suceder que los cambios sugeridos fueran demasiado caros o incluso imposibles de lograr con la plataforma de desarrollo escogida. Las recomendaciones de usabilidad sin embargo, representan los resultados de la prueba y deberán ser por lo tanto tomadas en cuenta por todos los involucrados en la usabilidad dentro del proceso de desarrollo.

Artefacto: Reporte general de usabilidad

Este documento es una descripción textual de cómo se desarrolló la prueba en términos generales. Este documento resultaría útil a cualquier persona interesada en saber los resultados de la prueba de usabilidad, desde los clientes y usuarios, hasta los líderes del proyecto y otros desarrolladores en general.

Como ya explicamos, el formato para reportar la prueba en general puede quedar a cargo del Analista de pruebas de usabilidad, la estructura que usamos en el caso de prueba (Capítulo 8) fue la siguiente (Ver Apéndice A, sección A.7):

- **Descripción de la prueba.** Incluye una breve descripción del sistema.
Escrito sin tecnicismos.
- **Validación** ¿Fue válida la prueba?
 - El sistema.
 - Los participantes.
 - * Monitores
 - * Usuarios.
 - * Equipo utilizado.
 - * Tareas
- **Resultados.**
 - Instrumento.
 - Calificación (Cuestionario de usabilidad).
- **Conclusiones.**

7.4 Trabajadores

Hemos propuesto una serie de artefactos necesarios para inyectar las pruebas de usabilidad al PU, con el propósito de dejar inalterado el proceso mismo. Los artefactos, como sabemos, son utilizados por los trabajadores en una actividad. A continuación expondremos a los trabajadores que utilizarán los artefactos que acabamos de mencionar.

7.4.1 Trabajador: Ingeniero de pruebas

El *Ingeniero de pruebas* (IP) existe ya en el PU y su trabajo consiste en planear todas las pruebas que se llevarán a cabo durante el último flujo de trabajo. Este trabajador realiza las actividades *Planificar prueba*, *Diseñar prueba* y *Evaluar prueba*.

Plantearemos la adición de dos actividades más: *Planear prueba de usabilidad* y *Diseñar prueba de usabilidad*. Esta actividad la explicaremos en la siguiente sección y será de la cual surgirán los artefactos que serán usados en la prueba misma.

7.4.2 Trabajador: Monitor de pruebas de usabilidad

El *Monitor de pruebas de usabilidad* (MPU) será el encargado de guiar a los usuarios en la prueba y tendrá mucho peso en el éxito de la misma.

En la elección de este trabajador se debe tener especial cuidado, no puede ser cualquier integrante del equipo de desarrollo. El monitor de pruebas de usabilidad debe tener, como ya lo explicamos en el Capítulo 5, cualidades especiales para lidiar con gente. Debe ser alguien que infunda confianza a los usuarios, alguien paciente que no ayude al usuario más de lo debido. Debe tener también buena dicción y presencia.

En la mayoría de los casos, el MPU tiene mucha influencia sobre el éxito o fracaso de la prueba: El monitor puede desesperar a los usuarios, hacerlos reír, hacerlos sentirse tranquilos, informarles adecuadamente sus tareas, ayudarles mal, etc. Es deseable entonces que el equipo de desarrollo capacite a alguien dentro del mismo equipo para que realice esta actividad

sistemáticamente. Lo ideal sería contratar a alguien especializado en tratar con gente, pero los costos de la prueba se verían incrementados.

7.4.3 Trabajador: Diseñador de interfaces de usuario

Este trabajador también existe también dentro del PU y es el que diseña la interfaz desde sus inicios. Esta actividad es llamada por el PU *Construir prototipo de la interfaz de usuario* se encuentra explicada en el Capítulo 6 y en [Jacobson et al, 2000].

Por ser este trabajador el que diseñó la IGU del sistema que se encuentra en construcción, es él mismo quien deberá estar presente en la prueba¹.

El diseñador de interfaces de usuario estará presente en la prueba y plasmará sus observaciones en las *Observaciones de prueba de usabilidad* que ya mencionamos. Deben ser los diseñadores de la interfaz los que estén presentes ya que son ellos los que conocen mejor su prototipo y los que entenderían mejor en un momento dado las recomendaciones de usabilidad.

7.4.4 Trabajador: Analista de pruebas de usabilidad

Finalmente introduciremos en el PU un trabajador más, el *Analista de pruebas de usabilidad (APU)*. Esta persona será la encargada de procesar muchos de los artefactos generados por los trabajadores anteriores y, mediante el análisis de los mismos, producir los Resultados de prueba de usabilidad.

Su trabajo es también muy importante ya que de éste dependerá que en la siguiente iteración en la que se realicen pruebas de usabilidad, se tenga la información suficiente para bien llevarlas a cabo y para generar una mejor interfaz en todos aspectos.

7.5 Actividades

Una vez que hemos expuesto los artefactos y los trabajadores que tendrán una presencia en todo lo concerniente a las pruebas de usabilidad, ha llegado

¹Recordemos que puede haber varias personas dentro del equipo de desarrollo con el mismo nombramiento

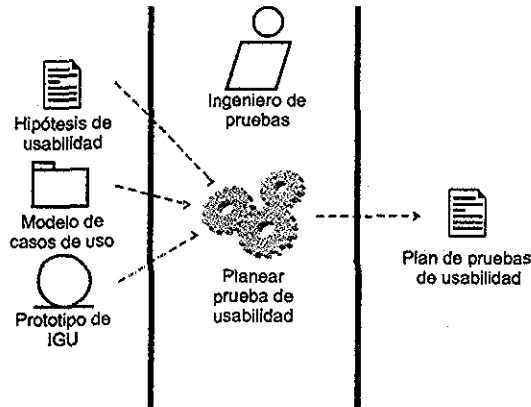


Figura 7.2: Entrada y resultado de la planeación de la prueba de usabilidad.

el momento de exponer y explicar las actividades que serán llevadas a cabo por los trabajadores que utilizarán los artefactos.

7.5.1 Actividad: Planear prueba de usabilidad

El IP planea la prueba de usabilidad. Como se muestra en la Figura 7.2, el ingeniero de pruebas, toma las hipótesis de usabilidad, el modelo de casos de uso y el prototipo de la interfaz, y produce un el *plan de pruebas de usabilidad*.

La forma de crear este artefacto no debe variar mucho ni entre pruebas dentro del mismo proyecto ni entre proyectos distintos. Se trata de sistematizar su creación para sistematizar así mismo las pruebas dentro del proceso de desarrollo.

El IP debe tener experiencia en pruebas de usabilidad y tiene, evidentemente, que tener una fuerte comunicación con el DIU para poder planear una prueba a la altura de los requerimientos de la misma

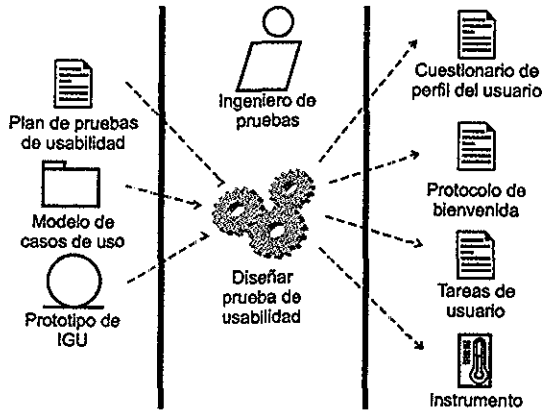


Figura 7.3: Entrada y resultado de la planeación de la prueba de usabilidad

7.5.2 Actividad: Diseñar prueba de usabilidad

En la Figura 7.3 se muestra el diagrama de actividades correspondiente. Para el protocolo de bienvenida, el cuestionario de perfil del usuario y el instrumento, se pueden confeccionar “machotes” para su uso en todas las pruebas. Las tareas de usuario sí serán específicas para las hipótesis suministradas, pero el proceso de creación de las mismas se deberá de mantener invariante.

Si bien los artefactos que se producen en esta actividad tienen como base el criterio y la experiencia del IP². De lo que se trata es de sistematizar su creación para reducir la variación en los resultados de las pruebas.

7.5.3 Actividad: Realizar prueba de usabilidad

Esta actividad (Figura 7.4) tiene como guía principal las tareas del usuario y el protocolo de bienvenida. El monitor de pruebas de usabilidad debe recibir al usuario y saludarlo amablemente antes de empezar la prueba.

²Aunque esto es cierto, las actividades y los artefactos que forman parte del proceso de prueba de la interfaz, se encuentran muy estudiados en IHC, reduciendo la capacidad de los trabajadores de hacer las cosas mal.

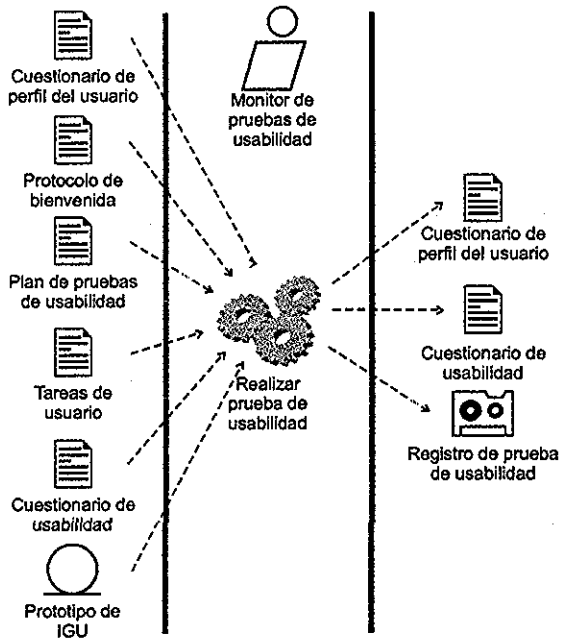


Figura 7.4: Entrada y resultado de guiar al usuario durante la prueba de usabilidad.

Una vez que ésta empieza, le debe de leer apropiadamente el protocolo de bienvenida y asegurarse que el usuario haya entendido todos los puntos. Posteriormente y con base en las tareas planeadas, el monitor debe indicarle al usuario qué actividades debe realizar y con qué material puede hacerlo.

Ejemplo:

Por favor capture los nombres de estos trabajadores (le da una ficha con los nombres) y de los de alta en la base de datos. Puede usted si lo desea, utilizar el manual del usuario que tiene a la derecha en el escritorio

El monitor puede recibir información por parte de los observadores (como explicamos en el Capítulo 5) a través de un dispositivo de comunicación a distancia. Es deseable que el usuario no se de cuenta que el monitor está recibiendo ayuda o instrucciones. Esta información puede ser ayuda en tiempo real de situaciones que no fueron previstas o bien sugerencias diversas.

Al finalizar la prueba, el monitor puede proveer al usuario del cuestionario de perfil del usuario (si no lo ha hecho antes) y, en su caso, del cuestionario de usabilidad. Debe darle las gracias y podrá aclararle aspectos de la prueba que no podían ser revelados *a priori*

7.5.4 Actividad: Observar prueba de usabilidad

Como ya mencionamos, esta actividad es llevada a cabo por el DIU. La prueba es observada en tiempo real por este trabajador (recordemos que se encuentra oculto en el laboratorio). En este sentido, esta actividad y la anterior pueden ser consideradas simultáneas, sin embargo, la creación de los artefactos en esta actividad se realiza post-prueba: Los observadores generarán las observaciones de la prueba de usabilidad posteriormente, utilizando los artefactos que se muestran en la Figura 7 5

El DIU debe tomar en cuenta toda la información contenida en los artefactos que utilizará para dar sus opiniones acerca de la prueba. El *Cuestionario de perfil del usuario* le dará información acerca de las capacidades y experiencia de los participantes, esto puede ser útil para ayudar discernir entre errores causados por la inexperiencia del usuario, por el sistema o por deficiencias en la prueba misma.

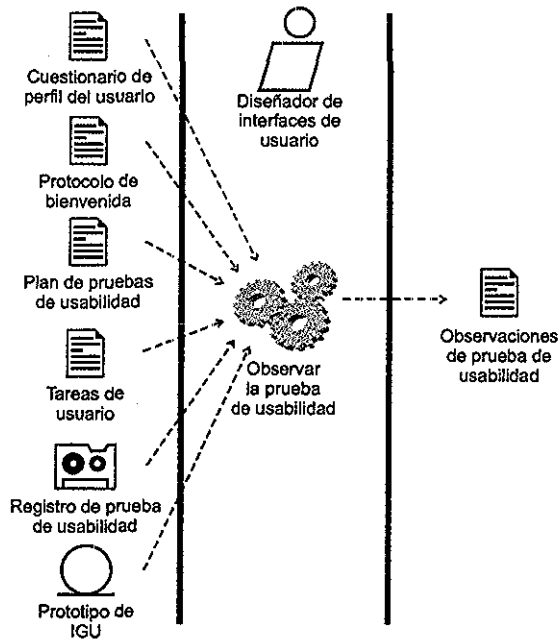


Figura 7.5: Entrada y resultado de observar la prueba de usabilidad.

El *Protocolo de bienvenida* revela exactamente lo que el usuario escuchó y es en general un documento primordial para emitir un juicio acerca de la prueba. Las *tareas de usuario* contienen todo lo que se le pidió al usuario así como la información que éste tenía a la mano, esto es necesario en caso de duda entre lo que el usuario hizo y lo que debió de haber hecho.

Recordemos que las *observaciones de la prueba de usabilidad* son sugerencias que no tienen que ser acatadas por los demás trabajadores involucrados en la prueba.

7.5.5 Actividad: Analizar prueba de usabilidad

Esta actividad (Figura 7.6) es importante, de ella depende en gran medida el éxito en lograr una interfaz realmente usable. Es difícil en el sentido de que utiliza más artefactos que ninguna otra y se tiene que reunir toda esta información en tres documentos (el paquete *Resultados de prueba de usabilidad*) que deben ser claros a muchos trabajadores e incluso, probablemente, a los clientes mismos.

Todos los artefactos que se utilizan y se muestran en la Figura 7.6 son importantes para reportar adecuadamente los resultados de la prueba. Evidentemente los cuestionarios informan al APU del perfil y las opiniones del usuario, el plan de pruebas revela las actividades que se le pidieron llevar a cabo.

El registro de la prueba es vital así como el prototipo. El APU debe tener el prototipo tal como se le presentó al usuario, debe poder ejecutarlo y explorarlo cuanto desee para así entender lo que sucedió en la prueba. Debe utilizar el instrumento para llevar a cabo las mediciones necesarias y debe leer cuidadosamente las Observaciones de prueba de usabilidad para conocer las opiniones del DIU acerca de la prueba misma: Recordemos que el DIU es el autor de la interfaz y sus opiniones acerca de la prueba son importantes.

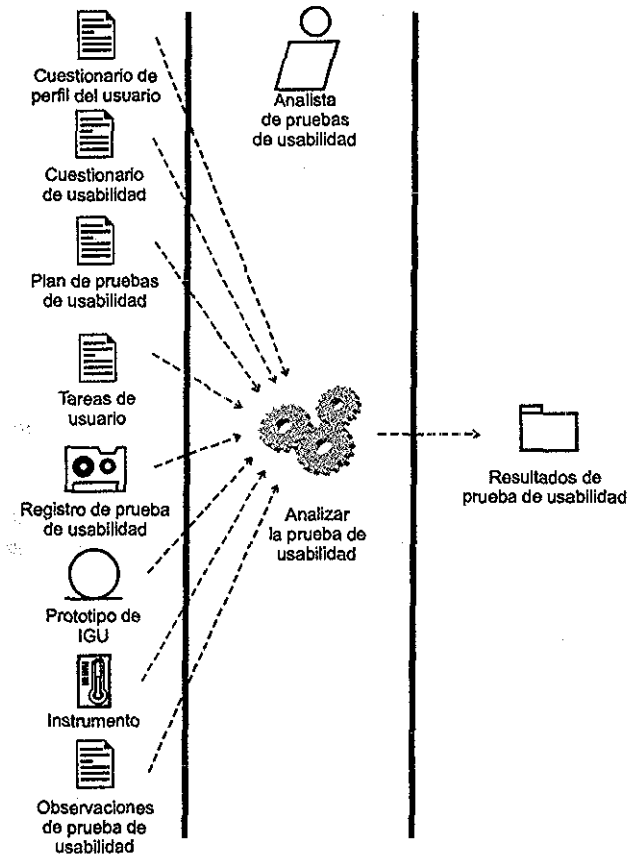


Figura 7 6: Entrada y resultado de realizar el análisis de la prueba de usabilidad

7.6 Diferencias entre flujos de trabajo

Sabemos que la carga de trabajo de los diferentes flujos no es la misma en las diferentes fases. Más aún, la forma de llevar a cabo las diferentes actividades puede variar entre fases. Para el caso de las pruebas de usabilidad esto es especialmente cierto.

Las actividades y el contenido de los artefactos que hemos presentado en las secciones anteriores tendrán pequeñas diferencias cuando se realicen o utilicen en diferentes fases del desarrollo. A continuación explicaremos estas diferencias.

7.6.1 Inicio y Elaboración

Como ya hemos mencionado, en estas fases es en las que se diseñará e implementarán los primeros prototipos de la futura IGU, en efecto, los profesionales de IHC establecen que la usabilidad se tiene que cuidar durante todo el proyecto [Nielsen, 1993]. Es por eso que, como lo mencionamos en este trabajo, nosotros planteamos una diferencia con el PU.

Mientras que el PU establece el primer diseño de la IGU en la fase de construcción, nosotros recomendamos realizar los diseños preliminares desde la fase de inicio si es posible, o a lo más tarde, en la fase de elaboración. Tomemos en cuenta que las pruebas de usabilidad pueden arrojar resultados que obliguen al equipo de desarrollo a contemplar nuevos requisitos que en algunos casos tienen un impacto importante en el desarrollo. Algunos riesgos, por lo tanto, son minimizados al realizar las pruebas de usabilidad en fases tempranas del proyecto.

Pero pensemos ahora que que estamos en la segunda iteración de la fase de inicio, muy probablemente el proyecto ya fue aprobado y buena parte de los requerimientos se tienen ya documentados. Adicionalmente se dispone de un primer bosquejo de cómo lucirá la interfaz y los diseñadores de la misma desean tener resultados tempranos de usabilidad.

Primero que nada se debe tener un prototipo que pueda ser probado. En caso de no contar con uno "ejecutable", se deben tener bosquejos detallados que sean capaces de proporcionar la información buscada. Como explicamos

en el Capítulo 5, se debe decidir entre una prueba horizontal o una vertical y generar el prototipo en "T" correspondiente.

La prueba de usuario junto con sus cuatro actividades tal y como la hemos descrito, se llevará a cabo. Recordemos sin embargo, que el levantamiento de los requerimientos no ha terminado y no se tienen en el prototipo, muchas de las características finales por lo tanto. Las primeras pruebas exploratorias serán entonces referentes a navegación y filosofía de uso en general. Conforme las pruebas exploratorias vayan avanzando en iteraciones posteriores de la fase de elaboración se podrán ir explorando partes más específicas del prototipo.

Es recomendable proporcionar al usuario desde estas pruebas un cuestionario de usabilidad como el SUS [Kirakowski], ya que el usuario podrá proporcionar información veráz aún si el prototipo está en un estado temprano

Así, en las pruebas de usabilidad que se efectúen en la fase de inicio habrá un artefacto que es posible que no se encuentre: El registro de prueba de usabilidad. Claramente si no es posible tener un prototipo ejecutable, podría no tener mucho sentido grabar las opiniones que los usuarios den de algunos bosquejos en papel. Para mayor información sobre las pruebas exploratorias ver el Capítulo 5.

7.6.2 Construcción

Cuando se pasa a la fase de construcción se debe tener ya un prototipo ejecutable y los resultados de pruebas anteriores habrán dado información suficiente para que la navegación, la filosofía de uso del sistema y buena parte de los elementos que aparecerán en pantalla estén ya implementados. Es tiempo entonces de hacer que el prototipo responda otro tipo de cuestiones.

Como explicamos ya en el Capítulo 5, las pruebas de diseño deben estar pensadas para contestar preguntas acerca de aspectos que van desde diseño gráfico hasta presentación de información y comandos. No debemos indagar sobre aspectos que debimos de conocer ya en las pruebas exploratorias, ya que el desarrollo del sistema podría sufrir serios retrasos si se regresara a este tipo de aspectos.

Todos los artefactos mencionados en la sección 7.3 pueden estar presentes. En este tipo de pruebas se hace más importante la existencia de un cuestionario de usabilidad ya que la interfaz debe tener un avance considerable en las últimas iteraciones de la fase de construcción.

7.6.3 Transición

Finalmente se tiene una interfaz que se debe de ver y debe funcionar muy similarmente a como será en su versión final, inclusive debe estar lista para proveer al usuario de una retroalimentación real, i.e, debe ser completamente funcional. Esto nos permitirá probar aspectos de la misma que tienen que ver con su desempeño

Supongamos que mediante los dos tipos de pruebas anteriores sabemos que nuestra IGU es fácil de usar, presenta información adecuada y pertinente, es atractiva e intuitiva, etc. Bien, ¿eso nos garantiza que el usuario de desempeñará eficientemente con ella? Más aún, ¿el usuario será más eficiente con este sistema que con el que tenía?

No es posible obtener mucha información sobre este tipo de cuestiones aplicando solamente los dos tipos de pruebas anteriores. Las pruebas de desempeño están diseñadas precisamente para hechar luz sobre aspectos de desempeño, eficiencia y tasa de errores de nuestra interfaz

En esta prueba deben utilizarse obligatoriamente todos los artefactos que hemos mencionado. Evidentemente algunos artefactos mantendrán esencialmente su contenido y otros no. En particular, las hipótesis de usabilidad y el plan de pruebas deberán tener un enfoque en el desempeño del sistema, recordemos que el prototipo será totalmente funcional y podremos, por lo tanto, ejercer presión sobre el mismo.

Si el sistema es un sistema distribuido que debe responder en tiempo real, la prueba de usabilidad incluso puede consistir en varios usuarios pidiendo recursos del sistema al mismo tiempo. En este tipo de pruebas es primordial simular los más parecidamente posible las condiciones reales en las cuales funcionará el sistema.

7.7 Invariancia del Proceso Unificado

El Proceso Unificado, como ya sabemos, es una metodología de desarrollo sólida resultado de mas de 30 años de investigación en desarrollo de software orientado a objetos. Actualmente el PU se considera un proceso sólido y efectivo, que ayuda a manejar la complejidad en el desarrollo de sistemas y que inova en la IS en muchos aspectos.

En medio de tantas bondades sin embargo, existe el hecho de que, al igual que muchas otras metodologías de desarrollo, la usabilidad y en particular la IGU no reciben la atención adecuada. Es por esto que decidimos complementar un poco al proceso para hacerlo un "mejor" proceso.

Sin embargo, por toda la solidez y efectividad que tiene el PU, la tarea debe llevarse a cabo con cuidado. En primer lugar, aunque es muy tentador, no se puede cambiar al PU *a placer*. La idea fue desde un principio que el PU permaneciera esencialmente invariante ante la propuesta. ¿Por qué? Porque estamos trabajando sobre la base de que el PU es un excelente proceso de desarrollo y por lo tanto su "filosofía" debe permanecer invariante.

Con "filosofía" nos referimos principalmente a la manera como ataca los problemas, su estructura interna, su visión de lo que debe ser un proceso de desarrollo de software. Esta "filosofía" tiene finalmente su correspondencia en la organización y la forma de pensar y trabajar del equipo de desarrollo. Esto es lo que queremos preservar.

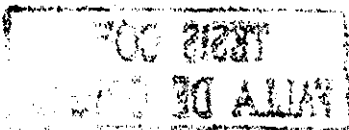
El reto es por lo tanto, proponer lo que nosotros consideramos *correcto* y suficiente *sin* modificar la esencia del PU. Ahora, si bien los cambios por sí solos pueden dar la impresión que se está cambiando sustancialmente al proceso, una visión panorámica del mismo antes y después de la propuesta nos permitirá dar una opinión más objetiva.

7.7.1 La estructura actual

En el PU veremos básicamente trabajadores, artefactos y actividades. Las actividades que se llevan a cabo definen la naturaleza del flujo de trabajo: No puede haber una actividad que suceda en dos flujos diferentes, sin embargo puede haber trabajadores que lleven a cabo actividades en más de un flujo

En la Figura 7.7 se muestra una gráfica que resulta especialmente ilustrativa. El eje del tiempo es el eje horizontal, por lo tanto las actividades que se encuentren más a la izquierda en la gráfica ocurrirán primero en el desarrollo. Resulta interesante observar que puede haber actividades de un flujo de trabajo que se lleven a cabo después que otras del siguiente flujo, i.e., las actividades entre flujos pueden invadirse en el tiempo.

Observemos que la Figura 7.7 nos proporciona bastante información acerca de la estructura del PU. Quién hace qué cosa, con qué lo hace y cuándo lo hace. Los cambios que haremos se verán reflejados claramente en esta figura, nuestro propósito es cambiarla lo menos posible y aún así proponer lo que creemos es necesario.



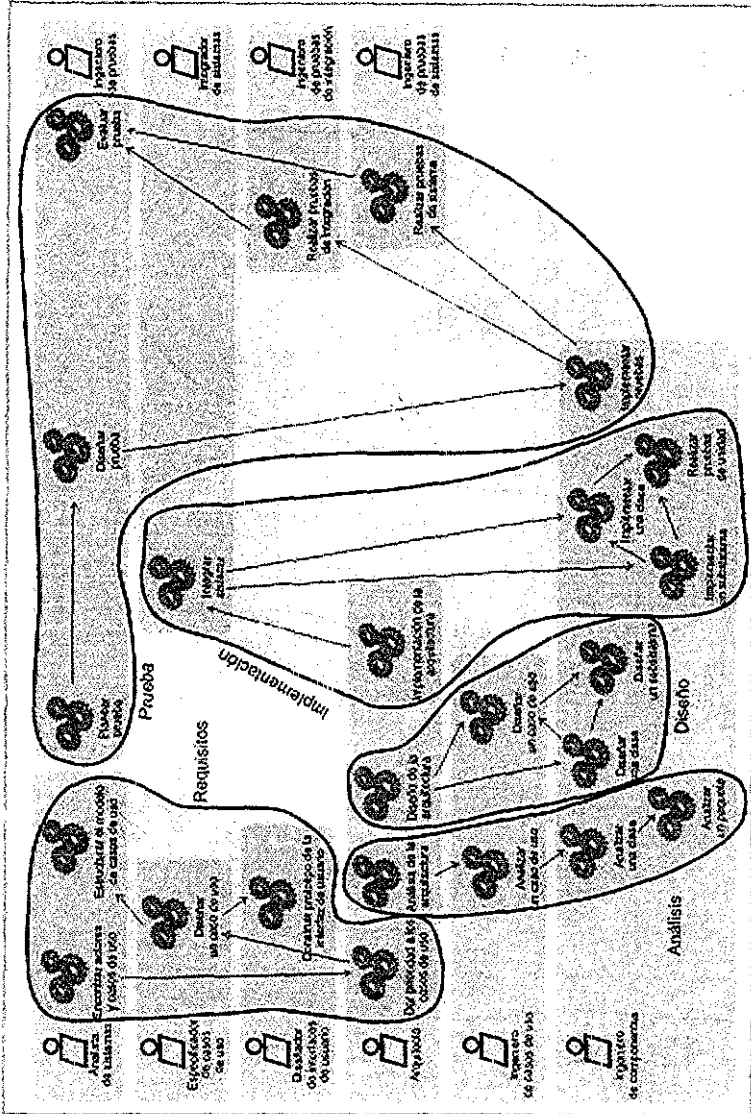


Figura 7.7: Estructura original del Proceso Unificado.

TESIS CON FALLA DE ORIGEN

7.7.2 La nueva estructura

Nuestra propuesta debe cambiar lo menos posible al PU y aún así ser lo más propositiva. Hemos agregado cinco actividades y dos trabajadores mas al proceso y debemos meterlos de manera que se respeten el tiempo y estructura general de las actividades. La Figura 7.8 muestra propuesta integrada ya al PU.

Creemos firmemente que el PU en esencia sigue siendo el mismo pero ahora contemplando adecuadamente las pruebas de usabilidad de manera que los desarrolladores que trabajan con el PU puedan adoptarlas sin cambiar sustancialmente su forma de trabajar, de pensar, de comunicarse, y obteniendo un producto cuya usabilidad se vió aumentada gracias a la pertinente aplicación de las pruebas de usabilidad adecuadas

Capítulo 8

Caso de prueba: Proyecto SIC

En el capítulo anterior expusimos una propuesta con la cual asegurábamos que se elevaría la usabilidad de un producto de software creado mediante el PU. Si esto se cumple o no, sólo se puede saber si se somete esta propuesta a prueba.

Las condiciones idóneas de aplicación son un tanto difíciles de lograr: Se necesita un proyecto de desarrollo que se lleve a cabo siguiendo al cien por ciento el Proceso Unificado de Desarrollo de Software, un proyecto en el cual se creen todos los artefactos que éste contempla y donde todos los trabajadores que deben existir realicen sus actividades respetando la estructura de la Figura 7.7. El proyecto debería de ser "tomado" desde el inicio para asegurar la correcta implementación de los tres tipos de prueba empezando con la exploratoria.

Se consiguió un proyecto de desarrollo cuyo equipo accedió a implementar la propuesta con el mayor rigor posible, i.e., se designó a los trabajadores y se llevaron a cabo las actividades produciendo los artefactos necesarios.

En este punto es prudente mencionar que el equipo no estaba utilizando al PU como metodología de desarrollo, sino TSP. Sin embargo se designaron a los trabajadores propuestos en el capítulo anterior y se produjeron todos y cada uno de los artefactos requeridos.

8.1 El proyecto

El proyecto de desarrollo consistió en una aplicación basada en formularios HTML Web, cuyo propósito era permitir a los investigadores de la UNAM dar de alta sus datos curriculares en una base de datos centralizada, así como realizar consultas y modificaciones sobre los mismos

La dependencia de la UNAM que ordenó el sistema fue la *Secretaría de Investigación y Desarrollo* (SID) Puesto que el objetivo era agilizar la captura del currículum y facilitar esta labor a los investigadores, una interfaz fácil de usar era deseable.

8.2 Las pruebas

En las siguientes subsecciones hablaremos de los tres elementos involucrados en las pruebas del sistema SIC: Participantes, laboratorio y trabajadores.

8.2.1 Los participantes

Ya que el sistema estaba pensado para proporcionar un servicio a los investigadores de la UNAM, nuestros usuarios tipo podían ser de dos tipos

1. Investigadores.
2. Asistentes o secretarías de investigadores.

Claramente muy pocos investigadores en la UNAM tienen acceso a una secretaria que llene su información curricular por ellos, pero muchos investigadores con puestos administrativos sí tienen esa posibilidad. Sin embargo, debido a su escasez, se pensó en utilizar exclusivamente investigadores para las pruebas.

La SID se comprometió a conseguir a los usuarios para la prueba. Como ya se mencionó en el Capítulo 5, para una prueba de usabilidad se necesitan por lo menos cuatro o cinco usuarios tipo; sin embargo, sólo se pudo conseguir a tres y una asistente de investigador: cuatro usuarios. Como la asistente efectivamente se encontraba entre los posibles usuarios tipo, decidimos incluirla. Como veremos en los resultados y en los artefactos de la prueba, fue

una decisión correcta ya que el desempeño del sistema con este usuario no fue muy diferente que con los otros tres.

8.2.2 El lugar

Las pruebas se llevaron a cabo en el *Laboratorio de interacción humano-máquina y desarrollo multimedia (IHM-MM)* del *Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET)* de la UNAM. Este laboratorio cuenta con las especificaciones suficientes descritas en el Capítulo 5 y mostradas en la Figura 5.2. En el laboratorio se generó el Registro de prueba de usabilidad, la Figura 8.1 muestra una imagen extraída de tal registro.



Figura 8.1: Imagen extraída del artefacto "Registro de prueba de usabilidad". En este caso se trató de un cassette VHS

8.2.3 Los trabajadores

El equipo de desarrollo del sistema se conformaba de tres programadores que hicieron también de diseñadores de interfaz de usuario. Por ser dueño de la propuesta, el Ingeniero de pruebas de usabilidad fui yo. Se coordinó al equipo

de desarrollo en cuanto a los artefactos a generar y la manera de generarlos, así mismo se planearon y diseñaron las pruebas de usabilidad.

Los monitores de pruebas de usabilidad los proporcionó el laboratorio de IHM-MM; fueron personas con experiencia en llevar a cabo este tipo de actividad.

Los diseñadores de la interfaz de usuario fueron los tres desarrolladores del sistema, tal y como ocurre en muchos procesos de desarrollo en los cuales los diseñadores de la interfaz forman parte del equipo de programadores. Como se hace evidente, este hecho entra en conflicto con las ideas y propuestas de esta tesis. Debido a que el proyecto fue tomado cuando el desarrollo se encontraba ya en etapa avanzada, fue imposible separar esta actividad.

De cualquier manera, la propuesta está pensada basada en la existencia de un trabajador especialmente capacitado en el diseño de interfaces y en el presente caso de prueba, esta no fue la realidad. Tomemos en cuenta, sin embargo, que existen muy pocas personas en el país realmente capacitadas para tal tarea, creemos que nuestra propuesta fomentaría la capacitación en diseño de interfaces en el equipo de desarrollo que escogiera utilizarla.

Un aspecto importante de las pruebas de usabilidad con usuarios es que permite al equipo de desarrollo mejorar la interfaz iterativamente y en la dirección correcta, lo cual hace que la falta de un verdadero experto en diseño de interfaces no afecte realmente la efectividad de la propuesta.

Finalmente, la única persona capacitada y disponible para llevar a cabo el análisis de la prueba fui yo, ya que había hecho ya algunos análisis de pruebas de usabilidad realizadas en el mismo laboratorio.

8.3 Los artefactos

8.3.1 Hipótesis de usabilidad

| | | | |
|-------------|-----------------------|------------|-----------------|
| Nombre | <u>Equipo TriSoft</u> | Fecha | <u>22/05/02</u> |
| Equipo | <u>TriSoft</u> | Instructor | <u>HO / GI</u> |
| Parte/Nivel | <u>Pruebas</u> | Ciclo | <u>1</u> |

Hipótesis y acciones de las pantallas para la 1ª evaluación del
"Sistema de Información Curricular"

Hipótesis generales

Sobre Usuarios:

- Los menús principales serán fácilmente comprendidos por el usuario.
- El usuario navegará sin dificultad entre los menús y las formas.
- El usuario siempre sabrá fácilmente en que menú esta y que esta haciendo.
- En cada menú el usuario sabrá que operaciones puede realizar (Altas y Consultas).
- Los colores no provocarán disgusto o molestia a los usuarios.
- Los mensajes mostrados al usuario serán fácilmente entendibles.
- Los mensajes de error indicarán al usuario en donde se equivocó y cuál es su error.
- El usuario siempre tendrá la opción de terminar la sesión cuando el dese.
- El usuario sabrá cuando sus datos han sido almacenados o modificados.
- Los textos del sitio deben de tener un tamaño suficiente para ser observados.
- El usuario no sabrá exactamente que contiene cada menú.

Sobre Sistema:

- Está completo en cuanto a accesos y vínculos pensados para la evaluación.
- No falla.
- Es semifuncional.

Figura 8.2: Página 1. Hipótesis de usabilidad. Proyecto SIC. Creado por el Diseñador de interfaz de usuario durante la actividad Prototipar interfaz de usuario. Usado por el Ingeniero de pruebas

| <u>Pantalla</u> | <u>Hipótesis</u> | <u>Acción</u> |
|----------------------------------|---|--|
| Login/Password | ... El usuario ... sabrá que debe proporcionar su login y password para acceder al sistema | 1. Entrar al sistema |
| Inicio | ... sabrá como entrar a los apartados de Datos Generales, Trayectoria Académica, Programas de Estímulos | 1. Describir en donde está 2. Pasar al apartado de Datos Generales y regresar |
| Menú Datos Generales | ... sabrá en donde se encuentra la sección de Datos Personales ... sabrá en donde se encuentra la sección de Direcciones | 1. Describir en donde está 2. Consultar los Datos Personales y regresar 3. Entrar a sección de Direcciones |
| Datos Personales Consulta | ... sabrá en donde se encuentra ... sabrá cómo editar la forma | 1. Editar los Datos Personales y regresar 2. Navegar a la sección de Direcciones |
| Datos Personales Edición | ... sabrá en donde se encuentra ... sabrá cómo almacenar la edición de la forma ... se percatará de los posibles errores que halla cometido en el llenado de su forma | 1. Almacenar la edición de los Datos Personales |
| Menú Direcciones | ... podrá fácilmente seleccionar que operación (Alta Consulta) desea realizar ... en caso de consulta se percatará de cómo recuperar los datos de una dirección específica | 1. Da de alta una dirección y regresar 2. Consulta una dirección |
| Direcciones Alta | ... sabrá en donde se encuentra ... sabrá cómo almacenar los datos de la forma ... se percatará de los posibles errores que halla cometido en el llenado de su forma | 1. Almacenar la Dirección |

Figura 8.3: Página 2. Hipótesis de usabilidad. Proyecto SIC

| | | | |
|-----------------------------------|--|---|--|
| <i>Direcciones Consulta</i> | ... sabrá en donde se encuentra | 1 | Editar las Direcciones |
| | ... sabrá cómo editar la forma | 2 | Navegar al apartado de Trayectoria Académica |
| <i>Direcciones Edición</i> | ... sabrá en donde se encuentra | 1 | Almacenar la edición de la Dirección |
| | ... sabrá cómo almacenar la edición de la forma ... se percatará de los posibles errores que halla cometido en el llenado de su forma | | |
| Menú Trayectoria Académica | ... sabrá en donde se encuentra la sección de Idiomas | 1 | Describir en donde está |
| | | 2 | Entrar a la sección de Idiomas y regresar |
| | | 3 | Entrar a la sección de Expertos Visitantes |
| <i>Menú Idiomas</i> | ... podrá fácilmente seleccionar que operación (Alta, Consulta) desea realizar | 1 | Da de alta un idioma y regresar |
| | ... en caso de consulta se percatará de cómo recuperar los datos de un idioma específico | 2 | Consulta un idioma |
| <i>Idiomas Alta</i> | ... sabrá en donde se encuentra | 1 | Almacenar dos idiomas |
| | ... sabrá cómo almacenar los datos de la forma ... se percatará de los posibles errores que halla cometido en el llenado de su forma | | |
| <i>Idiomas Consulta</i> | ... sabrá en donde se encuentra | 1 | Editar el Idioma |
| | ... sabrá cómo editar la forma | 2 | Navegar al apartado de Trayectoria Académica |
| <i>Idiomas Edición</i> | ... sabrá en donde se encuentra | 1 | Almacenar la edición del Idioma |
| | ... sabrá cómo almacenar la edición de la forma | | |

Figura 8.4: Página 3 Hipótesis de usabilidad. Proyecto SIC

| | | |
|------------------------------------|--|--|
| | ... se percatará de los posibles errores que halla cometido en el llenado de su forma | |
| <i>Menú Expertos Visitantes</i> | ... podrá fácilmente seleccionar que operación (Alta. Consulta) desea realizar | 1 Da de alta un Experto Visitante 2 Navegar al apartado de Programas de Estímulos |
| <i>Expertos Visitantes Alta</i> | sabrás en donde se encuentra ... sabrás cómo almacenar los datos de la forma ... se percatará de los posibles errores que halla cometido en el llenado de su forma | 1 Almacenar Experto Visitante |
| <i>Menú Programas de Estímulos</i> | ... podrá fácilmente seleccionar que operación (Alta. Consulta) desea realizar | 1 Da de alta un Programa de Estímulo 2 Sale del Sistema |
| <i>Pantalla de Salida</i> | ... se percatará de que su sesión a finalizado ... notará la presencia de una liga para acceder nuevamente al sistema | |

Figura 8 5: Página 4. Hipótesis de usabilidad. Proyecto SIC.

8.3.2 Plan de pruebas de usabilidad

| |
|-------------------------------------|
| Prueba: <u>SIC ciclo 1.ª Prueba</u> |
| Fecha: <u>26/05/2002</u> |
| Nota: _____ |

Plan de pruebas de usabilidad
Sistema de Información Curricular

1. Descripción del sistema
El sistema es accedido por Web y consta de código HIMI presentado dinámicamente mediante JSPs y JavaScript. Consta de formularios CGI mediante los cuales los investigadores de la UNAM podrán proporcionar sus datos curriculares y darlos de alta automáticamente en una base de datos centralizada. Así mismo, el sistema permite que dichos datos sean modificados y consultados por los mismos investigadores. Cada investigador tendrá un *login* y un *password* con el que podrá ingresar al sistema.
El prototipo tiene ya un alto grado de funcionalidad y diseño, por lo que la prueba de usabilidad cubrirá lo que normalmente se cubre en las pruebas exploratorias y de diseño. A pesar de que el sistema cuenta con una mayor funcionalidad, ésta será implementada en el segundo ciclo, por lo cual sólo la funcionalidad antes mencionada será probada.

2. Justificaciones
El equipo de desarrollo, y en particular los diseñadores de la interfaz, llegaron a la conclusión de que se necesitaba una prueba debido a que se llegó a un avance considerable en la implementación de la interfaz y existían dudas acerca del grado de usabilidad del mismo y de ciertas decisiones de diseño.
Debido a que se está cerca del final del primer ciclo de desarrollo es especialmente importante que las pruebas se realicen en este momento ya que se podrá entrar en el nuevo ciclo teniendo recomendaciones de usabilidad apropiadas.

3. Hipótesis de usabilidad para la prueba
Las siguientes hipótesis de usabilidad cubren las hipótesis de usabilidad entregadas por los diseñadores de interfaz de usuario. Estas han sido modificadas para poder generar más apropiadamente las actividades que se le pedirán al usuario durante la prueba.

Figura 8.6: Página 1 Plan de pruebas de usabilidad. Proyecto SIC. Creado y usado por el Ingeniero de pruebas durante la actividad Planear prueba de usabilidad

Hipótesis de usabilidad para la prueba:

- La filosofía general de uso del sistema será fácilmente entendida.
- Los menús no necesariamente son intuitivos.
- La información es fácilmente legible
- Los colores escogidos no son molestos o dificultan la usabilidad del sistema
- Los mensajes que proporciona el sistema son pertinentes y claros.
- El sistema informará oportunamente del éxito de las operaciones

Las actividades a pedírsele al usuario deben cumplir con el objetivo de informar sobre la veracidad de estas hipótesis.

Los diseñadores de la interfaz manifestaron su interés en que se probaran tres formularios "tipo", mismos que representan la funcionalidad de los demás. Las actividades a pedírsele al usuario deberán cubrir los siguientes tres formularios

- Dirección particular.
- Expertos invitados.
- Líneas de investigación

4. Requerimientos especiales para la prueba.

La prueba de usabilidad será una prueba "clásica", se realizará en el laboratorio del que se dispone y especialmente se necesitarán los requisitos siguientes:

- Debido a razones técnicas los datos básicos de los usuarios que asistan deberán de estar previamente cargados en la base de datos.
- Computadora con conexión a Internet ya que al sistema se accede por Web.
- Una ficha con datos sobre un experto invitado (sólo se requiere una debido a que este formulario no da de alta los datos a la BD).

Cinco fichas con datos personales, una por cada investigador

Figura 8.7: Página 2. Plan de pruebas de usabilidad. Proyecto SIC

8.3.3 Cuestionario de perfil del usuario

Prueba: SIC ciclo 1 M. Prueba
 Fecha: 30/05/2007
 Nota: 1.00

Cuestionario de perfil del usuario

1. ¿Cuál es su nivel académico?
DOCTORADO
2. ¿Desde cuándo usa computadoras?
1976
3. ¿Con qué frecuencia las utiliza?
DIARIAMENTE
4. ¿Hace cuánto utiliza Internet?
10 AÑOS
5. ¿Con qué frecuencia?
DIARIAMENTE
6. ¿Para qué lo utiliza?
BASES DE INFORMACION, EDICION Y RECOLECCION DE DATOS
7. ¿Qué entiende por Internet?
SISTEMA DE INFORMACION Y ANALISIS UNIVERSAL A TRAVES DE REDES DE COMPUTO.
8. ¿Está familiarizado con capturas de información?
SI
9. ¿Ha capturado información a través de Internet?
SI

Figura 8.8: Página 1. Cuestionario de perfil del usuario. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad. Usado por el Monitor de pruebas de usabilidad durante la actividad Realizar prueba de usabilidad.

Prueba: SIC a do 11. Prueba
 Fecha: 30/10/2006
 Nota: 25 Usuarios

Cuestionario de perfil del usuario

1. ¿Cuáles es su nivel académico?
 Candidato a Doctor
2. ¿Desde cuándo usa computadoras?
 20 años
3. ¿Con qué frecuencia las utiliza?
 Diario
4. ¿Hace cuánto utiliza Internet?
 6 años
5. ¿Con qué frecuencia?
 Diario
6. ¿Para qué lo utiliza?
 Búsqueda de información
7. ¿Qué entiende por Internet?
 Sistema interactivo de computadoras para
 facilitar el intercambio de información
8. ¿Está familiarizado con capturas de información?
 Si
9. ¿Ha capturado información a través de Internet?
 Si

Figura 8.9: Página 2. Cuestionario de perfil del usuario Proyecto SIC.

Prueba: SIC ciclo 1 1a Prueba
 Fecha: 015/06/2002
 Nota: 3.º USUARIO

Questionario de perfil del usuario

1. ¿Cuál es su nivel académico?

 Ma C.
2. ¿Desde cuándo usa computadoras?

 1982
3. ¿Con qué frecuencia las utiliza?

 Mediana
4. ¿Hace cuánto utiliza Internet?

 1998
5. ¿Con qué frecuencia?

 Mediana
6. ¿Para qué lo utiliza?

 Búsquedas
7. ¿Qué entiende por Internet?

 Accesos en tiempo real a Bases de Información
8. ¿Está familiarizado con capturas de información?

 Mediana
9. ¿Ha capturado información a través de Internet?

 Si

Figura 8 10: Página 3. Questionario de perfil del usuario. Proyecto SIC.

Prueba: SIC ciclo I M. B. B. B.
 Fecha: 08/06/2002
 Nota: 40 Usonsu, Secretaria

Cuestionario de perfil del usuario

1. ¿Cuál es su nivel académico?
Asistente
2. ¿Desde cuándo usa computadoras?
desde 1989
3. ¿Con qué frecuencia las utiliza?
diariamente
4. ¿Hace cuánto utiliza Internet?
1993
5. ¿Con qué frecuencia?
de manera cotidiana
6. ¿Para qué lo utiliza?
Buscar información.
7. ¿Qué entiende por Internet?
Es la herramienta actual más importante en BD de información nacional e internacional.
8. ¿Está familiarizado con capturas de información?
Si
9. ¿Ha capturado información a través de Internet?
Si

Figura 8.11: Página 4. Cuestionario de perfil del usuario. Proyecto SIC

8.3.4 Protocolo de bienvenida

| |
|--|
| Prueba: <u>SIC ciclo 1 1ª Prueba</u> Fecha: <u>26/05/2002</u> Nota: _____ |
| Protocolo de bienvenida para prueba de usabilidad del proyecto SIC |
| <p>Buenos días, mi nombre es {por definir} y estaré con usted en esta sesión Permítame explicarle porqué está usted aquí.</p> <p><i>Estamos probando el prototipo de un sistema que permitirá que los investigadores proporcionen sus datos curriculares vía Internet así como realizar consultas y modificaciones sobre los mismos.</i></p> <p>Mediante el estudio de sus acciones trataremos de determinar los defectos o virtudes de este sistema en términos de la facilidad con que usted interactúe con el mismo. Es importante enfatizar que el que será evaluado es el prototipo y no usted, por ello le pedimos que actúe con naturalidad, pero sobre todo, que responda con honestidad.</p> <p>Debido a que no es un sistema final existe la posibilidad de que no funcione adecuadamente. por favor no se extrañe si el sistema hace algo inesperado.</p> <p>Se le presentará un prototipo y se le pedirá que realice algunas tareas típicas para las cuales está diseñado el sistema, la sesión consistirá en que usted las efectúe y describa en voz alta sus acciones así como cualquier opinión que tenga ya que esto será de mucho ayuda para nosotros.</p> <p>Una vez comenzada la sesión siéntase en total libertad de hacer cualquier pregunta, aunque no podré contestar algunas de ellas, ya que el objetivo es simular la situación real en la cual operará el sistema de manera autónoma</p> <p>{ ¿tiene alguna duda? }</p> |

Figura 8.12: Página 1. Protocolo de bienvenida. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad. Usado por el Monitor de pruebas de usabilidad durante la actividad Realizar prueba de usabilidad.

8.3.5 Tareas de usuario

Prueba: SIC_ciclo 1.1* Prueba
 Fecha: 26/05/2002
 Nota: _____

Áreas de usuario

Ir tachando las actividades realizadas con el fin de no perderse durante la prueba NO se realizará ninguna actividad si antes no ha concluido la anterior (a menos que se le instruya lo contrario).

Las letras en itálicas presentan información que sólo debe conocer el monitor.

Nota: En caso de que el usuario cometa cualquier error preguntar: *¿El sistema le informó con claridad qué fue lo que pasó?*

1. Antes que nada ¿sabe cuál es el propósito del sistema?
2. Por favor entre a la página del sistema, la dirección está escrita en la ficha #1
3. Apuntando con el mouse, describa lo que ve en la pantalla e indique si entiende todo o no.
4. Ingrese al sistema los datos necesarios están en la ficha #2
5. Apuntando con el mouse, describa lo que ve en la pantalla e indique para qué cree que es cada elemento que observa en ella y cuál es su función.
6. ¿Puede leer cómodamente todos los textos que ve en esta página? (si o no)
7. ¿Le agradan los colores?
8. ¿Es usted investigador?
 SI: Por favor ingrese en el sistema sus propias líneas de investigación.

Ingreso de líneas de investigación. Trayectoria académica

NO: Por favor ingrese en el sistema las líneas de investigación que se le proporcionan en la ficha #3.

En cuanto haga click sobre Trayectoria académica preguntar

9. ¿Entiende el menú de la derecha?
10. ¿Puede describir que significa cada opción?
11. ¿Puede describir que hará cada opción?
12. Por favor continúe con el ingreso de sus líneas de investigación. No olvide indicar siempre qué está haciendo y en qué parte del sistema se encuentra.

Figura 8.13: Página 1 Tareas de usuario. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad Usado por el Monitor de pruebas de usabilidad durante la actividad Realizar prueba de usabilidad.

Cuando lo haya logrado preguntar:

- 1 ¿El sistema le avisó del éxito de la operación?
- 2 Regrese a la página principal
- 3 Por favor ingrese sus datos personales. Si lo desea, puede utilizar la información que se encuentra en la ficha #4

Ingreso a datos personales: Datos generales

Cuando lo haya logrado preguntar:

- 4 ¿El sistema le avisó del éxito de la operación?
- 5 Ahora, por favor ingrese al sistema un experto visitante. Si lo desea puede utilizar la información que se encuentra en la ficha #5

Ingreso a Expertos visitantes Trayectoria académica

Cuando lo haya logrado preguntar:

- 6 ¿El sistema le avisó del éxito de la operación?
- 7 Por favor salga del sistema

Las siguientes fichas deberán ser impresas aparte para que el usuario pueda utilizarlas separadamente de este documento durante la prueba

Ficha 1

http://diana.sid.unam.mx:8080/SICv1_ac/jsp/ver_1/index.jsp

Ficha 2

Login: inv

Password: inv

Figura 8.14: Página 2. Tareas de usuario. Proyecto SIC

| |
|---|
| <p>Ficha 3</p> <p>Microelección. Básica Media Intermetálicos. Aplicada Baja. 11 años de experiencia</p> <hr/> |
| <p>Ficha 4</p> <p>Acoxpa 56 Col. Rancho Viejo Ilalpan c p. 14390 Pais 1 invc@hotmail.com</p> <hr/> |
| <p>Ficha 5</p> <p>Smith Davis Berkley Pais 1 De noviembre de 1999 a agosto del 2001 Financiamiento: IBM Pais1</p> |

Figura 8 15: Página 3. Tareas de usuario. Proyecto SIC.

8.3.6 Instrumento

| Prueba: <u>SIC ciclo 1.ª Prueba</u> | |
|-------------------------------------|---|
| Fecha: <u>28/05/2002</u> | |
| Nota: _____ | |
| Instrumento | |
| Actividad del usuario | Tipo de resultados |
| 3 | <ul style="list-style-type: none"> • Número de elementos que entendió del total. |
| 4 | <ul style="list-style-type: none"> • SI/NO. • Número de errores. |
| 5 | <ul style="list-style-type: none"> • Número de elementos que entendió del total. |
| 6 | <ul style="list-style-type: none"> • SI/NO. |
| 7 | <ul style="list-style-type: none"> • SI/NO. |
| 9 | <ul style="list-style-type: none"> • SI/NO (qué entendió) |
| 10 | <ul style="list-style-type: none"> • SI/NO (opción por opción, en caso de NO ¿Qué entendió?) |
| 11 | <ul style="list-style-type: none"> • SI/NO (opción por opción, en caso de NO ¿Qué entendió?) |
| 12 | <ul style="list-style-type: none"> • SI/NO • Número de errores. • Tiempo • Comentarios. • ¿Hubo frustración? |
| 13 | <ul style="list-style-type: none"> • SI/NO |
| 14 | <ul style="list-style-type: none"> • SI/NO • Número de errores • Tiempo. • ¿Hubo frustración? |
| 15 | <ul style="list-style-type: none"> • SI/NO • Número de errores. • Tiempo. • ¿Hubo frustración? |
| 16 | <ul style="list-style-type: none"> • SI/NO |
| 17 | <ul style="list-style-type: none"> • SI/NO • Número de errores • Tiempo. • ¿Hubo frustración? |
| 18 | <ul style="list-style-type: none"> • SI/NO |
| 19 | <ul style="list-style-type: none"> • SI/NO • Número de errores. • Tiempo. • ¿Hubo frustración? |

Figura 8.16: Página 1. Instrumento. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad. Usado por el analista de pruebas de usabilidad durante la actividad analizar prueba de usabilidad.

8.3.7 Cuestionario de usabilidad

Prueba SIC ciclo 1^{ra} prueba.
 Fecha: 30/06/2002
 Nota: Muy buena. Opino que el sistema es estable y bueno.

Cuestionario de usabilidad

Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas.

| | Fuertemente en desacuerdo | | | | Fuertemente de acuerdo |
|---|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|
| 1. Creo que me gustará usar el sistema frecuentemente | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 2. Encuentro el sistema innecesariamente complejo | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 3. Pensé que el sistema era fácil de usar | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 5. Encontré que las distintas funciones del sistema estaban bien integradas | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 6. Pienso que había mucha inconsistencia en el sistema | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 7. Me imagino que la gente aprenderá a usar el sistema bastante rápido | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 8. Encuentro al sistema engorroso de usar | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 9. Me sentí muy seguro usando el sistema | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |
| 10. Necesito aprender muchas cosas antes de poder utilizar el sistema | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | 1 | 2 | 3 | 4 | 5 |

Figura 8 17: Página 1 Cuestionario de usabilidad. Proyecto SIC. Usado por el monitor de pruebas de usabilidad durante la actividad Realizar pruebas de usabilidad.

Prueba: SIC (sub 1) 1ª Prueba
 Fecha: 30/11/2006
 Nota: 7.50

Cuestionario de usabilidad

Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas.

| | Fuertemente en desacuerdo | | | | Fuertemente de acuerdo |
|---|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1. Creo que me gustará usar el sistema frecuentemente | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Encuentro el sistema innecesariamente complejo | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. Pensé que el sistema era fácil de usar | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Encontré que las distintas funciones del sistema estaban bien integradas | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. Pienso que había mucha inconsistencia en el sistema. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. Me imagino que la gente aprenderá a usar el sistema bastante rápido. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 8. Encuentro al sistema engorroso de usar | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. Me sentí muy seguro usando el sistema. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 10. Necesito aprender muchas cosas antes de poder utilizar el sistema | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Figura 8.18: Página 2 Cuestionario de usabilidad. Proyecto SIC.

Prueba: SIC ciclo 1 B Prueba
 Fecha: 5/10/2002
 Nota: 3er Vuelta

Cuestionario de usabilidad

Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas.

| | Fuertemente en desacuerdo | | | | | | Fuertemente de acuerdo |
|--|-------------------------------------|--------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|------------------------------|
| 1. Creo que me gustará usar el sistema frecuentemente. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | |
| 2. Encuentro el sistema innecesariamente complejo. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 3. Pensé que el sistema era fácil de usar. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| 4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 5. Encontré que las distintas funciones del sistema estaban bien integradas. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 6. Pienso que había mucha inconsistencia en el sistema. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 7. Me imagino que la gente aprenderá a usar el sistema bastante rápido. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 8. Encuentro al sistema oneroso de usar. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 9. Me sentí muy seguro usando el sistema. | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| 10. Necesito aprender muchas cosas antes de poder utilizar el sistema. | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Figura 8.19: Página 3 Cuestionario de usabilidad. Proyecto SIC.

TESIS CON FALLA DE ORIGEN

Prueba: SIC ciclo 1 15 Brecha
 Fecha: 05/06/2002
 Nom: 300 Vázquez

Cuestionario de usabilidad

Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas.

| | Fuertemente en desacuerdo | | | | Fuertemente de acuerdo |
|---|---------------------------------|---|---|---|------------------------------|
| 1. Creo que me gustará usar el sistema frecuentemente | 1 | 2 | 3 | 4 | 5 |
| 2. Encontré el sistema innecesariamente complejo | 1 | 2 | 3 | 4 | 5 |
| 3. Pensé que el sistema era fácil de usar | 1 | 2 | 3 | 4 | 5 |
| 4. Creo que necesitaré la ayuda de un técnico para poder usar el sistema | 1 | 2 | 3 | 4 | 5 |
| 5. Encontré que las distintas funciones del sistema estaban bien integradas | 1 | 2 | 3 | 4 | 5 |
| 6. Pienso que había mucha inconsistencia en el sistema | 1 | 2 | 3 | 4 | 5 |
| 7. Me imagino que la gente aprenderá a usar el sistema bastante rápido | 1 | 2 | 3 | 4 | 5 |
| 8. Encontré al sistema engorroso de usar | 1 | 2 | 3 | 4 | 5 |
| 9. Me sentí muy seguro usando el sistema | 1 | 2 | 3 | 4 | 5 |
| 10. Necesito aprender muchas cosas antes de poder utilizar el sistema | 1 | 2 | 3 | 4 | 5 |

Figura 8 20: Página 3 Cuestionario de usabilidad. Proyecto SIC

8.3.8 Observaciones de usabilidad

Prueba: Proyecto SIC

Fecha: 15/06/2002

Nota:

Observaciones de usabilidad

Primer Observador

Desde mi punto de vista particular, la prueba fue llevada a cabo de una manera exitosa.

Existieron algunos puntos que a mi me hubieran gustado cubrir también, pero por la premura del tiempo de prueba fue imposible, me refiero a las ediciones y consultas.

En cuanto a cada una de las pruebas, y específicamente hablando de los monitores siento que actuaron bien, la primera monitor tuvo una interacción menos personal con el usuario, cosa que me pareció magnífica, sin embargo, no seguía muy detalladamente la secuencia de las tareas. La segunda monitor interactuó posiblemente ayudando a los usuarios, pero seguía mas fielmente el guión de tareas a realizar. Esto podría solucionarse desde mi punto de vista, empleando un poco de tiempo previo a la prueba, para capacitar a los monitores.

Siento que lo ideal hubiera sido probar con los usuarios tipo, es decir únicamente investigadores en este caso, posiblemente después sería ideal probar con usuarios de apoyo como secretarías pero únicamente de este tipo. En esta prueba se mezcló un poquito el perfil de los usuarios.

En general, la prueba siento que es muy útil para cubrir los aspectos de usabilidad que son importantes dentro del sistema SIC.

Segundo Observador

Las pruebas para mí se realizaron exitosamente, lo único que siento que faltó de probar fue la edición y la consulta.

En cuanto a los usuarios con los que se hicieron las pruebas pienso que todos debieron ser investigadores y que fueran 5 investigadores, para tener una muestra más general, pero por falta de tiempo no se pudieron realizar. Pienso que después se deberán de realizar las pruebas con otros usuarios, los que vaciaran los datos de los Investigadores en el sistema.

Sobre las personas que guiaron las pruebas, siento que lo hicieron bien, aunque la primera monitor se me hizo mejor, primero porque seguía la secuencia de las tareas y cuando el investigador tenía duda no les decía nada dejaba que ellos buscarán la forma de resolver el problema. La segunda monitor lo que no me gusto fue que cuando veía que el usuario no podía resolver el problema, ella les decía directamente como hacerlo, y creo que en las pruebas eso no es conveniente. También creo que cuando se realicen las pruebas debe ser el mismo monitor.

Las pruebas realizadas anteriormente serán de mucha utilidad, ya que nos servirán para mejorar los aspectos de usabilidad de nuestro sistema para que sea más fácil para los usuarios de comprender y navegar.

Figura 8.21: Página 1. Observaciones de prueba de usabilidad. Proyecto SIC Creado por el Diseñador de interfaz de usuario y usado por el Analista de pruebas de usabilidad en la actividad Analizar prueba de usabilidad.

Tercer observador

Pienso que la prueba fue llevada a cabo correctamente, las asesorías por parte del Dr Fernando Gamboa y el Físico Rodrigo Pinzón fueron puntos clave para el logro de las pruebas.

La prueba realizada contó con un muy buen laboratorio de pruebas(Centro de Ciencias Aplicadas y Desarrollo Tecnológico). Allí se tiene una infraestructura muy sofisticada para poder llevar a cabo la grabación de pruebas, que fue la actividad central en las pruebas de usabilidad

En particular los monitores elegidos creo que fueron buenos, aunque definitivamente preferí la interacción lograda con el primero. Posiblemente esto se deba un poco a la premura con que fueron realizadas las pruebas en las que participó el segundo monitor

Me hubiera gustado haber efectuado las pruebas con los cinco usuarios, pues la metodología lo sugiere, aunque aquí fueron cuestiones fuera de nuestro alcance.

Figura 8.22: Página 2. Observaciones de prueba de usabilidad. Proyecto SIC.

8.3.9 Resultados de usabilidad

| |
|--|
| Prueba: <u>Proyecto SIC</u> Fecha: <u>15/06/2002</u> Nota: _____ |
| Reporte general de usabilidad |
| <p>1. Descripción de la prueba. El sistema a ser probado consistió en una aplicación Web que permitía a los investigadores de la UNAM dar de alta, consultar y modificar sus datos curriculares mediante formularios HTML.</p> <p>La necesidad de la prueba surgió cuando el proyecto se encontraba ya muy cerca de la terminación de la primera versión, lo cual obligó a planear una prueba que proporcionara la información que normalmente proporcionan las pruebas de prototipo y las pruebas de diseño. Por esto, las hipótesis de usabilidad proporcionadas por los diseñadores de la interfaz de usuario contemplaban tanto decisiones de navegación y estructura en general como decisiones de diseño específicas.</p> <p>El sistema a ser probado fue modificado para ofrecer sólo la funcionalidad a ser requerida. Se escogieron tres formularios representativos de la filosofía de uso general del sistema:</p> <ol style="list-style-type: none"> 1. "Dirección particular". 2. "Expertos visitantes". 3. Líneas de investigación". <p>Se probó el sistema con cuatro usuarios, tres investigadores de la UNAM y una secretaria. A todos los usuarios se les leyó el mismo protocolo de bienvenida y se les pidieron las mismas actividades a realizar</p> |
| <p>2. Validación de la prueba. Las pruebas de usabilidad llevadas a cabo resultaron ser exitosas en el sentido de que se obtuvo la información requerida y manifestada en las "hipótesis de usabilidad". Dos de tres observadores calificaron las pruebas como útiles para ellos y dijeron que les ayudaron a mejorar la calidad de su interfaz. Todos los observadores consideraron que las pruebas se llevaron a cabo "correctamente".</p> <p>2.1. El sistema El sistema fue perfectamente capaz de realizar las tareas que se diseñaron a partir de las hipótesis de usabilidad proporcionadas por los diseñadores de la interfaz. Así mismo, el sistema restringía el acceso sólo a aquellas áreas en las cuales tenían que entrar los usuarios. La única capacidad que el sistema no presentaba fue la de dar de alta los datos capturados a la base de datos, esta funcionalidad, sin embargo, sólo será requerida en el caso de ser realizada una prueba de desempeño, por lo que no es considerada importante. En resumen, el sistema fue el requerido, consistente con las tareas de usuario y las hipótesis de usabilidad.</p> <p>2.2. Los participantes El propósito principal del sistema es servir a los investigadores para dar de alta y consultar sus datos curriculares en la base de datos de la Secretaría de Investigación y Desarrollo. Por esta razón los usuarios tipo son investigadores de la UNAM. En esta prueba se</p> |

Figura 8.23: Página 1 Reporte de usabilidad. Proyecto SIC Creado por el Analista de pruebas de usabilidad, útil a todo interesado en la usabilidad del sistema y especialmente al Diseñador de interfaces de usuario en la siguiente iteración.

tuvieron a 4 de los cinco usuarios prometidos por la Secretaría de Investigación y Desarrollo, siendo uno de ellos una secretaria. Esto no es tan grave si se considera que algunos investigadores pueden disponer de una secretaria que dé de alta su curriculum. En este sentido la secretaria de un investigador es también un usuario tipo para este sistema.

La homogeneidad de los usuarios es algo descabido ya que las pruebas arrojan resultados igualmente homogéneos. Debido al bajo número de investigadores con los que se contaron (tres), la inclusión de la secretaria fue prácticamente obligatoria. El desempeño de la misma, sin embargo fue similar al de los investigadores, por lo cual los resultados de su prueba con el sistema serán considerados igualmente.

Todos los observadores de la prueba manifestaron en las "observaciones de prueba de usabilidad" su deseo de haber contado con más participantes así como de que los mismos tuvieran un perfil homogéneo. La disponibilidad de los usuarios fue algo que el cliente arregló y no fue posible mejorar.

2.3. Resultados del cuestionario de perfil de usuario.

Nivel académico

Tres de los cuatro participantes tenían un nivel académico superior a maestría. Uno de ellos reportó ser asistente.

Experiencia con computadoras

Todos los participantes reportaron una experiencia de más de diez años, todos ellos utilizan computadoras diariamente y generalmente para buscar información. Todos ellos entienden la naturaleza de Internet.

Experiencia específica para uso del sistema

Todos los participantes reportaron estar familiarizados con el concepto de captura de información y lo han hecho a través de Internet.

2.4. Las tareas

El artefacto "tareas de usuario", se confeccionó lo más específicamente posible. Se tuvo la intención de que el monitor tuviera un guión que no diera lugar a equivocaciones o ambigüedades. Se agarraron las hipótesis de usabilidad originales y se resumieron, de ahí salieron diecinueve actividades que cubrían estas hipótesis y guiaban cuidadosamente al monitor en su actividad. El registro de "pruebas de usabilidad" mostró entendimiento de las tareas tanto por parte de los usuarios como por parte del monitor. Dos de los tres observadores manifestaron su deseo de haber tenido más tiempo para poder cubrir más partes del sistema.

2.5. Los monitores

Los monitores que se escogieron fueron individuos con algo de experiencia en realizar este tipo de actividades. Antes de comenzar la prueba los monitores leyeron las "tareas de usuario" así como el protocolo de bienvenida y se aseguraron junto con el Ingeniero de Pruebas de haber comprendido bien los documentos.

Los monitores en algunos momentos mostraron inexperiencia para manejar a los usuarios y negar ayuda. A pesar de esto, los tres observadores calificaron a los monitores como "buenos" pero sugirieron una mejor capacitación de los mismos e hicieron notar que, en las dos últimas pruebas, los monitores pudieron haber ayudado en algunos momentos al usuario. Uno de los observadores manifestó su deseo de tener homogeneidad en los monitores.

Figura 8.24: Página 2. Reporte de usabilidad. Proyecto SIC.

2.6. Otros aspectos
El hardware utilizado en la pruebas fue elegido correctamente, esto se puede concluir ya que ninguno de los participantes se quejó al respecto o mostró tener problemas debido a esto. El laboratorio utilizado también resultó adecuado, se tuvo un registro de la prueba claro y útil como fuente de información. Uno de los observadores se mostró especialmente contento con el desempeño del mismo.

3. Resultados
El sistema recibió opiniones muy favorables. Esto queda avalado por la opiniones de los participantes durante y después de la prueba, esto está registrado tanto en el "registro de prueba de usabilidad" como en la calificación que cada uno le puso al sistema y que se conoce gracias al "cuestionario de usabilidad"

3.1. Análisis de las pruebas
A continuación mostramos el análisis detallado de las pruebas de usabilidad para cada uno de los cuatro usuarios. Las actividades se encuentran en el artefacto "Tareas de usuario".

Prueba: SIC ciclo 1, 1ª prueba
Fecha: 30/05/2002
Nota: Primer usuario

| Actividad del usuario | Tipo de resultados |
|-----------------------|---|
| 3 | Número de elementos que entendió del total. Todos los elementos (3 de 3). |
| 4 | SI/NO. SI Número de errores 0 Errores |
| 5 | Número de elementos que entendió del total. 10 de 11 (elementos de menú superior y botón de terminar sesión). |
| 6 | SI/NO. NO (letras borrosas, tamaño pequeño). |
| 7 | SI/NO NO |
| 9 | SI/NO (qué entendió) SI <i>No estuvo de acuerdo en el orden en el que se muestran las opciones.</i> |
| 10 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) SI. Describió todas las opciones |
| 11 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) SI. Describió todas las opciones. |

Figura 8.25: Página 3 Reporte de usabilidad. Proyecto SIC.

| | |
|----|---|
| 12 | <p>SI/NO SI Número de errores. 0 Errores Tiempo. 5:34 min. Comentarios. Para el usuario "prioridad" no tiene sentido, no entendió bien para que o que indica prioridad ¿Hubo frustración? NO</p> |
| 13 | <p>SI/NO SI</p> |
| 14 | <p>SI/NO SI Número de errores 0 Errores Tiempo. 0:20 min. ¿Hubo frustración? NO Comentarios. Se debería de cambiar el orden de los menús en "trayectoria académica" (Usuario)</p> |
| 15 | <p>SI/NO SI Número de errores. 1 error No se percató de los campos marcados con "" Tiempo. 5:09 min. ¿Hubo frustración? NO Comentarios En "estado", aceptó números (Sistema). Sugirió que en "teléfono" se aceptaran letras, no entendió la etiqueta de "sólo dígitos".</p> |
| 16 | <p>SI/NO SI</p> |
| 17 | <p>SI/NO SI Número de errores. 1 error en el sistema, no acepto el punto en el campo Nombre de la institución. Tiempo. 2:30 min. ¿Hubo frustración? NO.</p> |

Figura 8.26: Página 4 Reporte de usabilidad. Proyecto SIC

| 18 | SI/NO SI <i>Le marcó error en el campo de "Nombre de la Institución".</i> |
|---|--|
| 19 | SI/NO SI Número de errores 6 Errores. Tiempo. 0:51 min. ¿Hubo frustración? SI Comentarios <i>No notó la presencia de la liga de salida.</i> |
| Prueba: SIC ciclo 1, 1ª prueba Fecha: 30/05/2002 Nota: Segundo usuario. | |
| Actividad del usuario | Tipo de resultados |
| 3 | Número de elementos que entendió del total 3 de 3. |
| 4 | SI/NO. SI Número de errores. 0 Errores |
| 5 | Número de elementos que entendió del total. 10 de 11 (elementos del menú superior y botón de terminar sesión). |
| 6 | SI/NO. SI Comentarios. Dijo que tenía el contraste adecuado. Se confundió con la opción de "Inicio" que queda activa en rojo y al pasar el mouse sobre las demás opciones se ponen del mismo color. |
| 7 | SI/NO SI |
| 9 | SI/NO (qué entendió) SI Comentarios <i>Mencionó nuevamente el orden en el que están dispuestos los elementos del menú. "Cómputo" podría ser nombrado como "herramientas de cómputo" (Usuario). Habría que marcar la diferencia en el color de los títulos del menú (Usuario).</i> |
| 10 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) SI |

Figura 8.27: Página 5. Reporte de usabilidad. Proyecto SIC

| | |
|----|---|
| 11 | <p>SI/NO (opción por opción, en caso de NO ¿Qué entendió?)</p> <p>SI</p> <p>Comentarios</p> <p>Los títulos del Menú tienen los mismos colores que las ligas, eso le causó un poco de confusión.</p> <p>En el menú principal la opción inicio siempre está en rojo, eso lo desconcertó con los otros menús, porque así tiene dos opciones en rojo.</p> <p>Sugirió que se llame "Manejo de Idiomas" en lugar de "Idiomas" y "Manejo" o "Dominio de Herramientas de cómputo" en lugar de "Cómputo".</p> |
| 12 | <p>SI/NO</p> <p>SI</p> <p>Número de errores.</p> <p>0 Errores.</p> <p>Tiempo.</p> <p>3:35 min.</p> <p>Comentarios.</p> <p><i>Le desconcertó que hubiera 5 líneas de investigación, creyó que debía llenar forzosamente las cinco líneas</i></p> <p>La información que se da para líneas de investigación no está en la descripción del menú "trayectoria académica".</p> <p>El orden de los menús le desconcertaron, dijo que después de "trayectoria académica" debería seguir "Trayectoria Laboral".</p> <p>Después de que llena la forma y la envía no sabe que acción tomar, no le "queda claro".</p> <p>¿Hubo frustración?</p> <p>NO</p> |
| 13 | <p>SI/NO</p> <p>SI</p> |
| 14 | <p>SI/NO</p> <p>SI</p> <p>Número de errores</p> <p>0 Errores.</p> <p>Tiempo.</p> <p>0:16 min.</p> <p>¿ Hubo frustración?</p> <p>NO</p> |
| 15 | <p>SI/NO</p> <p>SI</p> <p>Número de errores.</p> <p>0 Errores.</p> <p>Tiempo.</p> <p>1:27 min.</p> <p>¿ Hubo frustración?</p> <p>NO</p> |
| 16 | <p>SI/NO</p> <p>SI</p> |

Figura 8.28: Página 6. Reporte de usabilidad. Proyecto SIC.

| 17 | <p>SI/NO SI Número de errores. 1 error, no llenó el primer campo, no lo vió. Tiempo. 12:45 min. ¿Hubo frustración? SI Comentarios: No llegó fácilmente a "Expertos Visitantes", no se le recordó bien donde está. La información proporcionada fue general, no le ayudó mucho la descripción de los menús, faltó la mención de las tareas específicas. La barra que dice "Regresar a >>" le desconcertó. La información del campo en la pregunta "Se está realizando actualmente?" No la entendió con claridad. El orden de los campos en la forma le desconcertó. Dijo que el primer campo debería de llamar mas la atención, "que se resalte más ese campo". Comprendió bien que significan los asteriscos, pero sugirió que se hiciera mas énfasis, que fueran mas llamativos.</p> |
|---|---|
| 18 | <p>SI/NO SI</p> |
| 19 | <p>SI/NO SI Número de errores 0 Errores. Tiempo. 0:09 min. ¿ Hubo frustración? NO Comentarios Mencionó que después de un tiempo de interactuar con el sistema, se puede adaptar fácilmente a pesar de los errores o mal diseño.</p> |
| <p>Prueba: SIC ciclo 1, 1ª prueba Fecha: 05/06/2002 Nota: Tercer usuario.</p> | |
| Actividad del usuario | Tipo de resultados |
| 3 | Número de elementos que entendió del total 3 de 3. |
| 4 | SI/NO SI Número de errores. 0 Errores. |

Figura 8.29: Página 7. Reporte de usabilidad. Proyecto SIC

**TESIS CON
FALLA DE ORIGEN**

| | |
|----|---|
| 5 | Número de elementos que entendió del total 10 de 11 <i>No encontró desarrollos tecnológicos</i> |
| 6 | SI/NO NO Comentario Las letras son muy chicas. |
| 7 | SI/NO SI Comentario Debería de cambiarse el contraste. |
| 9 | SI/NO (qué entendió) SI Comentario No entendió "Intercambio Académico", "Expertos Visitantes" y "Participación Académica". Buscó ayuda en línea en el sistema. Mencionó que sería bueno implementar un tipo de ayuda. Se confundió con la etiqueta "regresar a" del menú de ubicación en el sistema, trató de dar click sobre este varias veces. |
| 10 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) SI. Comentario El usuario entendió todos los menús excepto "Intercambio Académico". No entiende nada, ni sabe nada sobre Intercambio Académico, no sabe que es "Experto Visitante". |
| 11 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) NO Comentarios: El usuario supo que hacer en todos las opciones, excepto en "Expertos Visitante" y "Participación Académica", no le queda claro que puede hacer en cada uno de ellos. |
| 12 | SI/NO SI Número de errores 4 Errores. Tiempo 2:53 min. Comentarios En el campo prioridad tuvo duda. Propuso que debería haber un botón de regreso y ponerlo en un lugar donde se viera siempre. No encontró fácilmente la forma "líneas de investigación". Tuvo muchos problemas en regresar. Propuso que debería de haber una ayuda, para cuando se tenga duda de un campo o una liga. ¿Hubo frustración? SI |

Figura 8 30: Página 8 Reporte de usabilidad. Proyecto SIC

| | |
|---|--|
| 13 | SI/NO SI |
| 14 | SI/NO SI Número de errores 0 Errores. Tiempo. 0:05 min. ¿ Hubo frustración? NO |
| 15 | SI/NO SI Número de errores 0 Errores Tiempo. 2:37 min. ¿ Hubo frustración? NO |
| 16 | SI/NO SI |
| 17 | SI/NO SI Número de errores. 1 error. Lo cometí cuando lleno el campo de fecha Tiempo. 3:00 min ¿Hubo frustración? NO. |
| 18 | SI/NO SI |
| 19 | SI/NO SI Número de errores 0 Errores Tiempo. 0:19 min. ¿ Hubo frustración? NO |
| Prueba: SIC ciclo 1, 1ª prueba Fecha: 05/06/2002 Nota: Cuarto usuario (Asistente de investigador) | |
| Actividad del usuario | Tipo de resultados |
| 3 | Número de elementos que entendió del total 3 de 3 |

Figura 8.31: Página 9. Reporte de usabilidad. Proyecto SIC

| | |
|----|---|
| 4 | SI/NO SI Número de errores. 0 Errores. |
| 5 | Número de elementos que entendió del total. 10 de 11(elementos de menú superior y botón de terminar sesión) Comentarios <i>Mencionó que podría ponerse "experiencia laboral" en lugar de "trayectoria laboral" y "Productividad Científica" en lugar de "Productividad Académica". No supo que era "presentación de curriculum".</i> |
| 6 | SI/NO NO Comentarios "Las letras son muy chicas", mencionó que se aumentarían unos 2 puntos más. |
| 7 | SI/NO NO Comentarios Le gustaría que en el menú de encabezado fuera utilizado un rojo más fuerte que no se pierda con el fondo amarillo. |
| 9 | SI/NO (qué entendió) SI Comentarios Mencionó que podría cambiarse la organización del menú, "superación académica" al final. |
| 10 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) NO. Comentarios El usuario entendió todos los menús excepto "Cómputo" y "Participación Académica". |
| 11 | SI/NO (opción por opción, en caso de NO ¿Qué entendió?) NO Comentarios del usuario: El usuario supo que hacer en todas las opciones, excepto en "Participación Académica". Sugirió ordenar en el menú "trayectoria académica" que algunas ligas debieran ir antes que otras. |
| 12 | SI/NO SI Número de errores. 0 Errores Tiempo. 0:51 min. ¿Hubo frustración? NO |
| 13 | SI/NO SI |

Figura 8.32: Página 10. Reporte de usabilidad. Proyecto SIC

| | |
|----|---|
| 14 | <p>SI/NO SI Número de errores 0 Errores Tiempo. 0:46 min. ¿ Hubo frustración? NO</p> |
| 15 | <p>SI/NO SI Número de errores 2 errores <i>No pudo capturar con éxito en los primeros dos intentos el teléfono particular, ingresaba guiones.</i> <i>No lograba editar el teléfono.</i> Tiempo. 9:08 min ¿ Hubo frustración? SI Comentarios El usuario tuvo problemas con el campo: En "Tipo de dirección", no supo bien como seleccionar la opción particular. No entendía a que refería el campo "país". Cuando ve le mensaje de error no supo que hacer. Para modificar el error se confundió y oprimió el botón modificar en lugar de posicionarse en el campo que le estaba marcando el error.</p> |
| 16 | <p>SI/NO SI</p> |
| 17 | <p>SI/NO SI Número de errores. 3 errores. El campo tipo de estancia lo dejo vacío. Tiempo. 02:55 min ¿Hubo frustración? NO.</p> |
| 18 | <p>SI/NO SI Comentarios El monitor le ayudó a encontrar la opción de expertos visitantes.</p> |
| 19 | <p>SI/NO SI Número de errores 0 Errores Tiempo. 0:05 min. ¿ Hubo frustración? NO</p> |

Figura 8 33: Página 11. Reporte de usabilidad. Proyecto SIC

3.2. Calificación del sistema

A continuación se presenta la calificación obtenida por el sistema. La calificación se obtuvo aplicando el cuestionario SUS (System Usability Scale). Este cuestionario reporta un aproximado de la usabilidad del sistema en términos de las impresiones inmediatas de los participantes justo después de la prueba.

Cada participante emitió una calificación y todas ellas se promediaron para dar la calificación final

| | |
|-----------|------|
| Usuario 1 | 90 |
| Usuario 2 | 80 |
| Usuario 3 | 90 |
| Usuario 4 | 87.5 |

Calificación final: 86.8

El cuestionario divide las opiniones de los participantes en cinco posibilidades, si la escala va de 0 a 100 y la dividimos en cinco:

muy malo, malo, regular, bueno, muy bueno

podemos concluir que a los usuarios, el sistema les pareció "muy bueno", lo cual coincide fuertemente con las opiniones recibidas después de la prueba. Mediante ciclos de usabilidad prueba-implementación-prueba se espera un ligero aumento en la calificación final del sistema

4. Conclusiones

Como todo sistema, y más aún, como todo sistema en construcción con una fuerte interacción con el usuario, éste presenta errores de diseño en la interfaz humana. A pesar de esto, todos los participantes mostraron capacidad para usarlo y entenderlo. Cabe recordar que las pruebas se enfocaron en tres tipos de formularios considerados representativos de la filosofía general de uso del sistema, por lo tanto, los resultados que mostramos en este documento, serán considerados como representativos del sistema completo.

Los usuarios opinaron que el sistema era básicamente un "muy buen sistema". Las calificaciones que éstos pusieron al mismo coinciden con esta opinión

A pesar de que el sistema mostró su capacidad de ser usado fácilmente, existen características del mismo que tienen que ser corregidas para minimizar la tasa de errores y aumentar el nivel de satisfacción de los usuarios. Estas correcciones se hacen patentes mediante una revisión experta de la interfaz y mediante el análisis del registro videográfico de las pruebas.

Figura 8.34: Página 12. Reporte de usabilidad. Proyecto SIC.

| Actividad del usuario | Tipo de resultados |
|-----------------------|---|
| 3 | <p>Usuario 1: Entendió todos los elementos.</p> <p>Usuario 2: Entendió todos los elementos.</p> <p>Usuario 3: Entendió todos los elementos.</p> <p>Usuario 4: Entendió todos los elementos.</p> |
| 4 | <p>Usuario 1: Si, 0 errores.</p> <p>Usuario 2: Si, 0 errores.</p> <p>Usuario 3: Si, 0 errores.</p> <p>Usuario 4: Si, 0 errores.</p> |
| 5 | <p>Usuario 1: Entendió 10 de 11 elementos.</p> <p>Usuario 2: Entendió 10 de 11 elementos.</p> <p>Usuario 3: Entendió 10 de 11 elementos.</p> <p>Usuario 4: Entendió 10 de 11 elementos.</p> |
| 6 | <p>Usuario 1: No leyó bien las letras.</p> <p>Usuario 2: Leyó bien las letras.</p> <p>Usuario 3: No leyó bien las letras.</p> <p>Usuario 4: No leyó bien las letras.</p> |
| 7 | <p>Usuario 1: No.</p> <p>Usuario 2: Si.</p> <p>Usuario 3: Si.</p> <p>Usuario 4: No.</p> |
| 9 | <p>Usuario 1: Si.</p> <p>Usuario 2: Si.</p> <p>Usuario 3: Si.</p> <p>Usuario 4: Si.</p> |
| 10 | <p>Usuario 1: Si.</p> <p>Usuario 2: Si.</p> <p>Usuario 3: Si.</p> <p>Usuario 4: No.</p> |
| 11 | <p>Usuario 1: Si.</p> <p>Usuario 2: Si.</p> <p>Usuario 3: No.</p> <p>Usuario 4: No.</p> |
| 12 | <p>Usuario 1: Si, 0 errores, 5:34 min, no hubo frustración.</p> <p>Usuario 2: Si, 0 errores, 3:35 min, no hubo frustración.</p> <p>Usuario 3: Si, 4 errores, 2:53 min, hubo frustración.</p> <p>Usuario 4: Si, 0 errores, 0:51 min, no hubo frustración.</p> |
| 13 | <p>Usuario 1: Si.</p> <p>Usuario 2: Si.</p> <p>Usuario 3: Si.</p> <p>Usuario 4: Si.</p> |
| 14 | <p>Usuario 1: Si, 0 errores, 0:20 min, no hubo frustración.</p> <p>Usuario 2: Si, 0 errores, 0:16 min, no hubo frustración.</p> <p>Usuario 3: Si, 0 errores, 0:05 min, no hubo frustración.</p> <p>Usuario 4: Si, 0 errores, 0:46 min, no hubo frustración.</p> |

Figura 8.35: Página 1. Resumen de los resultados obtenidos de los usuarios.

| | |
|----|--|
| 15 | Usuario 1: Si, 1 error, 5:09 min, no hubo frustración. Usuario 2: Si, 0 errores, 1:27 min, no hubo frustración Usuario 3: Si, 0 errores, 2:37 min, no hubo frustración Usuario 4: Si, 2 errores, 9:08 min, hubo frustración. |
| 16 | Usuario 1: Si Usuario 2: Si Usuario 3: Si. Usuario 4: Si. |
| 17 | Usuario 1: Si, 1 error, 2:30 min, no hubo frustración Usuario 2: Si, 1 error, 12:45 min, hubo frustración. Usuario 3: Si, 1 error, 3:00 min, no hubo frustración. Usuario 4: Si, 3 errores, 2:55 min, no hubo frustración |
| 18 | Usuario 1: Si. Usuario 2: Si. Usuario 3: Si Usuario 4: Si. |
| 19 | Usuario 1: Si, 6 errores, 0:51 min, hubo frustración. Usuario 2: Si, 0 errores, 0:09 min, no hubo frustración. Usuario 3: Si, 0 errores, 0:19 min, no hubo frustración Usuario 4: Si, 0 errores, 0:05 min, no hubo frustración. |

Figura 8.36: Página 2 Resumen de los resultados obtenidos de los usuarios.

Algunas recomendaciones de usabilidad se apoyan o se valen de criterios ergonómicos específicos [Bastien, 1993]

| |
|---|
| Prueba: <u>Proyecto SIC</u> Fecha: <u>15/06/2002</u> Nota: _____ |
| Recomendaciones de usabilidad |
| <p>El presente documento presenta una serie de recomendaciones de usabilidad para el sistema SIC. Estas recomendaciones se construyen tanto a partir de los resultados de las pruebas de usabilidad efectuadas al final del primer ciclo de desarrollo del sistema SIC, como a partir de una revisión a criterio del analista de pruebas de usabilidad.</p> |
| <p>Generales</p> <ul style="list-style-type: none"> • Las ligas que indican la navegación no fueron fácilmente entendidas. Mejorar la manera en la cual se puede regresar a pantallas anteriores. Sugerencia: Dejar siempre el menú de la izquierda visible. Criterio de <i>Densidad de información</i>. • Tanto el formato de la liga a "terminar sesión" como su leyenda no se entienden, hacer que resalte y cambiar leyenda a "Salir del sistema" o similar. Viola el criterio de <i>Legibilidad</i>. |
| <p>Pantalla de inicio</p> <ol style="list-style-type: none"> 1 Cambiar las palabras "login" y "password" por "clave de usuario" y "contraseña" o similares en español por consistencia con los sistemas Web existentes en este país. Viola el criterio de <i>Compatibilidad</i>. |
| <p>Pantalla del menú principal</p> <ol style="list-style-type: none"> 2 Cambiar el formato de presentación de la liga "terminar sesión" de manera que el usuario lo identifique inequívoca y fácilmente. Criterio de <i>Legibilidad</i>. 3. Quitar el texto "Regresar a >> inicio" resulta innecesario y no fue identificado por ningún usuario. Criterio de <i>Agrupación y distinción de elementos</i>. 4. Aumentar el tamaño de las letras de las ligas principales de forma que se puedan leer cómodamente. Criterio de <i>Legibilidad</i>. 5. A criterio de los desarrolladores y a sugerencia de algunos usuarios cambiar ligeramente los colores de manera que resalten más las letras. Criterio de <i>Legibilidad</i>. |
| <p>Menú izquierdo de "trayectoria académica"</p> <ul style="list-style-type: none"> • Cambiar el azul de los títulos del menú debido a que son confundidos por ligas. Criterio de <i>Legibilidad</i>. • Sugerencia: Cambiar el orden de los elementos del menú de manera que resulten lógicos. Los cuatro usuarios se quejaron del orden de estos elementos. Criterio de <i>Agrupación y distinción de elementos</i>. <p>Sugerencia: Dejar el menú siempre a la vista para facilitar la navegación. Criterio de <i>Densidad de información</i>.</p> |

Figura 8.37: Página 1. Recomendaciones de usabilidad. Proyecto SIC.

Forma de "líneas de investigación"

1. Agregar las líneas de investigación al texto que acompaña trayectoria académica" en la pantalla principal. Criterio de *Aviso*.
2. Mostrar una etiqueta que explique brevemente "Prioridad" Criterio de *Aviso*.
3. No mostrar todos los campos de líneas de investigación de primera vez, ir mostrándolos dinámicamente conforme se vayan llenando. El usuario se confunde y piensa que se tienen que llenar todos. Criterio de *Aviso*.
4. Cuando se graba el formulario de líneas de investigación, proporcionar al usuario una manera sencilla de continuar con otra actividad o llevarlo al inicio. Uno de los usuarios mostró confusión al no saber qué hacer después. Criterio de *Retroalimentación inmediata*.

Forma de "líneas de investigación"

- Por guía de estilo los radjans deben estar pre-seleccionados con alguna opción por defecto. Criterio de *Compatibilidad*.
- Poner los "*" con algún color que resalte. Criterio de *Legibilidad*.
- El campo "Estado" debe estar en un combo ya que múltiples errores pueden ocurrir como se vio en las pruebas. Criterio de *Protección contra errores*.
- Cambiar la etiqueta "Sólo dígitos" por "Sólo números" Criterio de *Significado de códigos*.
- El botón "Modificar" resulta ambiguo cuando se trata de corregir un error, cambiar su nombre por "Aceptar" o similar. Criterio de *Significado de códigos*.

Forma de "expertos visitantes"

- Preseleccionar los "radjans" con un valor por defecto. Criterio de *Compatibilidad*.
- Los campos de nombre de la institución deben aceptar puntos. Se sugiere un combo. Criterio de *Protección contra errores*.
- El concepto de expertos visitantes causa confusión. Lo mismo ocurre con la mayoría de los campos que se presentan. Forma especialmente problemática. se sugieren etiquetas de explicación. Criterio de *Aviso*.
- Resaltar los "*". Criterio de *Legibilidad*.

Finalmente se mencionará que el número de recomendaciones no representa la usabilidad del sistema probado. En este caso el sistema recibió resultados muy buenos en términos de usabilidad.

Figura 8.38: Página 2. Recomendaciones de usabilidad. Proyecto SIC

8.4 Resultados

Estrictamente, los resultados de las pruebas de usabilidad son todos los artefactos presentados en este capítulo, sin embargo, estos resultados ejercen una influencia que pasa al equipo de desarrollo y puede llegar incluso a ejercer influencias importantes sobre los clientes

Llevar a cabo las pruebas de usabilidad tiene siempre un costo. Para un proyecto de la vida real, este costo es tanto monetario como temporal y tiene que ser absorbido tanto por el cliente como por los desarrolladores. En este caso en particular, el costo fue de tiempo, y el equipo de desarrollo se vió en la necesidad de explicar que parte de ese tiempo fue invertido en realizar investigaciones para aumentar la usabilidad del sistema y por lo tanto de la calidad global del mismo

Bien, parte de lo que buscamos con la propuesta es que los resultados de las pruebas de usabilidad puedan ser reinyectados de nuevo al proceso de desarrollo como parte de los nuevos requerimientos al inicio de una nueva iteración¹.

Bien, el equipo de desarrollo tardó en poder arrancar el nuevo ciclo² ya que se levantaron nuevos requerimientos debido a que los resultados de las pruebas hicieron que los clientes se replantearan ciertos aspectos del sistema.

Para conocer la opinión que el equipo de desarrollo tuvo acerca de varios aspectos de la parcial aplicación de la propuesta, se creó un cuestionario para ser contestado por el mismo equipo. En las Figura 8.35 y 8.36 se muestra una tabla con las preguntas efectuadas así como con las respuestas de los tres integrantes del equipo y sus comentarios.

¹En este caso hablamos de ciclos de desarrollo.

²Donde en particular se realizarían las nuevas implementaciones a la interfaz.

| | |
|---|--|
| <p>1 ¿Qué tanto les sirvieron las pruebas de usabilidad?</p> <p>a. nada. b. muy poco. c. poco. d. regular. e. mucho.</p> | <p>Mucho.</p> <p>Mucho. Aunque me hubiera gustado que fueran más extendidas.</p> <p>Mucho.</p> |
| <p>2 ¿Qué tan difícil resultaron de implementar?</p> <p>a. nada. b. muy poco c. poco. d. regular. e. mucho.</p> | <p>Muy poco.</p> <p>Poco. Aunque los cambios mapeados a la documentación son algo largos.</p> <p>Regular.</p> |
| <p>3 ¿Qué tanto sienten que las pruebas perturbaron la estructura del proyecto?</p> <p>a. nada. b. muy poco. c. poco. d. regular e. mucho</p> | <p>Nada.</p> <p>Regular. Aunque las pruebas no generaron cambios significativos, despertaron interés por parte de los clientes de hacer modificaciones.</p> <p>Regular. No se tenían muy bien calendarizadas dentro del proyecto usando TSPi.</p> |
| <p>4 ¿Desde su punto de vista, el proyecto</p> <p>a. empeoró? b. no cambió? c. mejoró?</p> | <p>Mejóro.</p> <p>Mejóro.</p> <p>Mejóro. Indudablemente mejoró, indicaron mejoras contundentes en aspectos que pasábamos por alto.</p> |
| <p>5 ¿Creen que finalmente las pruebas</p> <p>a. perjudicarán a los usuarios? b. ni los perjudicarán ni los beneficiarán? c. los beneficiarán?</p> | <p>Los beneficiarán.</p> <p>Los beneficiarán. Estoy muy de acuerdo con eso.</p> <p>Los beneficiarán.</p> |
| <p>6 ¿Creen que después de las pruebas los usuarios</p> <p>a. opinarán mejor de la interfaz? b. opinarán igual? c. opinarán peor?</p> | <p>Opinarán mejor.</p> <p>Opinarán mejor.</p> <p>Opinarán mejor.</p> |

Figura 8.39: Página 1 Cuestionario para evaluación de los resultados de las pruebas. Proyecto SIC.

| | |
|--|---|
| <p>1. ¿Creen que al cliente le gustará más el sistema?</p> <p>a. SI.</p> <p>b. NO.</p> | <p>SI.</p> <p>Después de las pruebas, además de los cambios propios a la interfaz, el cliente sugirió diversos cambios en cuando a la semántica, obviamente al realizar estos cambios es fácil suponer que el cliente debe quedar satisfecho. En pocas palabras, por lo menos las pruebas le abrieron los ojos al cliente para preocuparse más por la calidad y la supervisión de su propio proyecto.</p> <p>SI. Solo que en este punto opino que hay que educar al cliente sobre el requerimiento implícito que es la usabilidad. Es probable que mientras él no lo entienda, descuide la importancia de éste aspecto.</p> |
| <p>2. ¿Consideran que el incorporar las pruebas podría traer beneficios para el cliente?</p> <p>a. SI.</p> <p>b. NO.</p> | <p>SI.</p> <p>SI. Eso es seguro.</p> <p>SI.</p> |
| <p>3. Con las pruebas ¿Creen que la usabilidad del sistema</p> <p>a. aumentará?</p> <p>b. será igual?</p> <p>c. disminuirá?</p> | <p>Aumentará.</p> <p>Aumentará.</p> <p>Aumentará.</p> |
| <p>4. ¿Creen que el proceso de implementación del sistema</p> <p>a. se dificultó?</p> <p>b. mantuvo su nivel?</p> <p>c. se simplificó?</p> | <p>Mantuvo su nivel.</p> <p>Se dificultó. Supongo que si se hubiera seguido un plan desde el principio donde se contemplaran las pruebas en las fases importantes como requerimientos y diseño, se hubieran simplificado muchas cosas. En nuestro caso, las pruebas fueron casi finales, dado lo cual no fue un test sobre un prototipo semifuncional sino ya funcional, por lo cual si hay dificultad para realizar modificaciones a estas alturas pues impacta a documentos de otras fases.</p> <p>Mantuvo su nivel.</p> |

Figura 8 40: Página 2 Cuestionario para evaluación de los resultados de las pruebas. Proyecto SIC

14/9

TESIS CON
FALLA DE ORIGEN

Capítulo 9

Conclusiones

Como ingenieros de software nuestro interés debe ser siempre crear software de calidad (robusto, mantenible, usable, etc.) guiado por un proceso de desarrollo que nos permita lograrlo, además de minimizar costos, terminar en tiempo y presupuesto además de bien documentado.

Propusimos al Proceso Unificado de Desarrollo de Software como un proceso que promete lograr lo que acabamos de mencionar además de unificar la manera en la cual se logra y se comunican los ingenieros de software entre sí.

No existe el proceso de desarrollo perfecto, siempre habrán nuevas ideas que mejoren lo ya existente. A este respecto, parte del trabajo de un ingeniero de software es siempre buscar la manera de hacer mejor su trabajo: Crear sistemas de software de calidad.

En ese orden de ideas, mejorar los procesos de desarrollo en busca del software ideal, a partir de incorporar la usabilidad en los mismos, es inmediato. En este sentido resalta el hecho de que a pesar de ser algo nuevo, no se siente la presencia de esta idea entre los procesos de desarrollo más populares a pesar de que sí es un tema de mucha importancia entre los profesionales que se encuentran relacionados con la IHC.

Sentimos entonces que el proponer lo expuesto en este documento es justo, necesario e importante. Hay que hacer notar que la propuesta no es incorporar la usabilidad dentro de los procesos de desarrollo, cosa que de hecho apoyamos fuertemente. La propuesta es simplemente la siguiente.

Incorporar la forma de llevar a cabo pruebas de usabilidad dentro del Proceso Unificado, para así aumentar esta característica en el software producido, y esto, sin cambiar la esencia del mismo.

Probar el éxito de la propuesta requiere de ser aplicada en desarrollos que sobrepasan el tiempo y el alcance de esta tesis. La propuesta se aplicó y refinó en varios desarrollos, siendo en SIC donde pudo mostrarse con mayor detalle y fidelidad. La factibilidad de la misma dependía de que se cumplieran los siguientes puntos

- La propuesta debía ser una propuesta útil, implementable y bien fundamentada.
- La propuesta debía poder ser implementada sin perturbar importante-mente al Proceso Unificado
- Los artefactos debían ser útiles a las pruebas, a los desarrolladores y contribuir a la buena documentación del proyecto.
- Los resultados debían ser fácilmente re-inyectables al proceso.

9.1 Las pruebas de usabilidad

Como ya se explicó, la razón de incluir el caso de prueba en este trabajo fue probar la factibilidad de la propuesta; i.e, probar que la propuesta era implantable dentro de un proyecto de desarrollo con Proceso Unificado de manera que se cambiara lo menos posible al mismo.

A partir del caso de prueba presentado, podemos decir que se logró implantar en un proceso de desarrollo basado en TSPi con toda la estructura planeada: artefactos, trabajadores, actividades. Esta estructura fue especialmente diseñada para acoplarse al PU.

Podemos agregar que, como lo expresaron los integrantes del equipo de desarrollo, la propuesta benefició al proyecto y dió muestras de que aportaría beneficios también para el cliente y los usuarios. Evidentemente el beneficio para los desarrolladores se desprende de los anteriores: Los desarrolladores siempre resultan beneficiados de cualquier logro relacionado con el proyecto.

9.1.1 Los resultados

La estructura de la propuesta estuvo siempre pensada en el contexto de PU. La Figura 7.8 (pág. 102) muestra el lugar temporal que ocupa en el contexto para el cual fue pensada. Los trabajadores y las actividades específicas que realizan están fuertemente ligados al PU. Los artefactos sin embargo, son documentos que pueden ser generados en cualquier proceso de desarrollo, el éxito y la utilidad de los mismos no está atada a algún proceso específico.

El que un equipo de desarrollo pudiera llevar a cabo exitosamente la propuesta constituye en sí una primera prueba de la *factibilidad* de la misma. Creemos firmemente que el implantarla en un contexto de PU será más sencillo y presentará menos problemas (si es que los hubo en esta primera aproximación), ya que para eso está diseñada.

Por lo que respecta a los resultados de las pruebas, bien, merecen ser remarcados, 1) las adiciones pudieron ser inyectadas en un proyecto de desarrollo real con todos y cada uno de los artefactos requeridos, las actividades y los trabajadores que las llevaron a cabo. Estos artefactos mostraron su utilidad en cada parte de las pruebas y más aún para el análisis de las mismas.

2) Estas pruebas mostraron ser de suma importancia y utilidad al proyecto en el cual fueron implementadas, esto tiene relación con el hecho de que el sentimiento general de todos los relacionados con ellas, incluyendo los usuarios tipo, fue que las pruebas les parecían sumamente útiles, esto nos hace pensar que el proyecto fue un mejor proyecto y el sistema final será un mejor sistema.

9.1.2 Evaluación de las pruebas

¿Cómo podemos evaluar el grado de éxito de las pruebas efectuadas? El objetivo de las pruebas de usabilidad con usuarios es aumentar la usabilidad del software producido. Si se aplican las pruebas y se encuentran errores cuya corrección mejorará la usabilidad de la interfaz, podemos hablar ya de un éxito de las mismas.

En el caso en cuestión, de la aplicación de las pruebas surgieron potenciales mejoras¹ a la interfaz que darán como resultado un aumento de la

¹A la fecha de la terminación de este trabajo, las mejoras todavía no eran terminadas

usabilidad.

Adicionalmente, de la información mostrada en las figuras 8.36 y 8.37 (pp 144, 145) podemos concluir los siguientes puntos en opinión de los desarrolladores del sistema probado:

- Las pruebas sirvieron mucho al proyecto.
- Fueron relativamente poco difíciles de implementar.
- Perturbaron relativamente poco el proyecto original (se agregó un ciclo para incorporar los cambios).
- El proyecto mejoró.
- Los futuros usuarios se beneficiarán y opinarán mejor de la interfaz.
- El cliente obtendrá beneficios, aunque un retraso en el desarrollo.
- Es muy probable que al cliente le guste más el sistema.
- La usabilidad aumentará.
- El proceso de implementación del sistema aumentó ligeramente de dificultad.

9.2 El panorama

Veamos todo esto en una perspectiva más amplia: Se generó una propuesta cuidadosamente pensada para ser aplicada en un contexto del PU y cuyo objetivo era complementar el proceso de desarrollo ya existente con la intención de proveer una manera de aumentar la usabilidad del futuro sistema sin perturbar el proceso de desarrollo mismo.

Se escogió un medio de implantación que nos permitiera verificar y evaluar la pertinencia y bondades del enfoque dentro del proceso. Si bien el contexto difirió en cuanto a la metodología escogida, la estructura de lo que se hizo fue exactamente la que se tenía planteada

de implementar.

Se sometió entonces a prueba la propuesta y se evaluó su efectividad y su factibilidad. Ambas características fueron favorables en opinión de los que la probaron y en opinión del creador de la misma.

9.3 Un mejor proceso

En la creación de esta propuesta trabajamos sobre la base de que el PU tal y como se encuentra actualmente es completo en todos menos uno de los aspectos involucrados en la creación de software: El aseguramiento de la usabilidad.

Nuestra propuesta fue concebida para ayudar a llenar este "hueco" logrando finalmente un mejor proceso. La justificación debe haber quedado suficientemente clara a estas alturas. Hemos podido verificar en una primera aproximación la utilidad y la pertinencia de las pruebas. Hemos también obtenido indicios de que la inclusión de las pruebas no tienen porqué ser difíciles de implementar y si proporcionan información muy útil al proyecto. Esto último está extensamente documentado en la literatura de IHC y fue claramente evidenciado en los resultados obtenidos en las pruebas realizadas.

A pesar de que no fue posible aplicar la propuesta al PU, por haber sido especialmente diseñada para el mismo, el acoplamiento dará aún mejores resultados que los logrados en el caso de prueba. Para verificar esto se tendrá que comprobar que lo siguiente ocurre:

- Si la propuesta puede ser introducida desde el inicio en un proyecto basado en PU, de forma que el calendario del proyecto, el costo, el volumen de documentación y el esfuerzo humano cambie sólo marginalmente.
- Si los resultados de cada prueba de usabilidad pueden ser reinyectados en la siguiente iteración como nuevos requerimientos no funcionales.
- Si la interfaz de usuario final refleja el resultado del primer diseño y de aquellos cambios fruto de las pruebas de usabilidad llevadas a cabo.

Entonces podremos asegurar que el proceso resultante es simplemente un mejor proceso. En el laboratorio de IHM-MM del CCADET se tienen planes para trabajos futuros con esta propuesta.

9.4 Conclusiones finales

El Proceso Unificado de Desarrollo de Software es una metodología probada y respaldada por décadas de investigación. La propuesta que presentamos fue cuidadosamente diseñada para respetar todo lo que es válido en este proceso y añadir sólo aquello que creemos necesita.

La filosofía y estructura originales fueron respetadas. Creemos que el PU sigue siendo básicamente el mismo pero con un "plus". Un plus que lo hace mejor y que cubre una necesidad existente en todo desarrollo de software que interactúa con seres humanos: El poder saber si las decisiones que se toman en la interfaz de usuario son correctas.

El objetivo del caso prueba fue principalmente sentir qué tan bien se desempeñaban los artefactos dentro de nuestra estructura de actividades, desde saber qué tan útiles resultan las hipótesis de usabilidad hasta conocer qué tan incorporables son los resultados de usabilidad a las siguientes iteraciones (ciclos en este caso). Adicionalmente buscábamos hablar de un caso concreto en el cual las pruebas de usabilidad hubieran probado su utilidad.

Creemos que presentamos una propuesta relativamente simple, bien adaptada al PU y cuyo objetivo está bien definido: Guiar al desarrollador que usa el PU para incorporar pruebas de usabilidad y aumentar la usabilidad de su producto. Creemos que la propuesta tiene lo que se necesita para ser probada en el contexto correcto y podremos esperar, con base en la experiencia previa, que se obtendrán beneficios para todos los involucrados en el proyecto de desarrollo.

Apéndice A

Artefactos de las pruebas

En el capítulo 8 se muestran los artefactos utilizados en las pruebas de usabilidad del proyecto SIC. Al realizar este trabajo nos imaginábamos a un equipo de desarrollo que, utilizando el PU, quisiera incorporar la usabilidad a su proyecto. Además del esfuerzo efectuado en detallar y exponer claramente la propuesta, queremos facilitar un poco más el entendimiento de la propuesta y su implementación mediante formatos base para la creación de los artefactos.

Algunos de los artefactos consisten simplemente en una lista de hipótesis o un texto breve con las observaciones textuales de la prueba. En este apéndice presentamos una propuesta de cómo podrían lucir algunos de los artefactos más complejos. Evidentemente estos formatos son sólo una propuesta y el equipo de desarrollo es libre de decidir el formato y el contenido de cada uno de ellos. En cualquier caso esperamos que resulten sumamente útiles e ilustrativos al lector.

A.1 Plan de pruebas de usabilidad

| |
|---------------|
| Prueba: _____ |
| Fecha: _____ |
| Nota: _____ |

Plan de pruebas de usabilidad
Nombre del sistema

1. Descripción del sistema.
Se describe brevemente el sistema en desarrollo. Los objetivos principales son darle un contexto a las hipótesis de usabilidad y proveer de un preámbulo a las justificaciones para la prueba.

2. Justificaciones.
El Ingeniero de pruebas de usabilidad deberá escribir aquí las razones por las cuales el equipo de desarrollo se decidió a llevar a cabo una prueba de usabilidad. Se deberá incluir también lo que los diseñadores de la interfaz esperan de la prueba, lo cual, evidentemente está fuertemente ligado a las justificaciones.

3. Hipótesis de usabilidad para la prueba.
Se incluyen en forma de lista las hipótesis de usabilidad (concretas y concisas) que el Ingeniero de pruebas de usabilidad extrajo de las hipótesis de usabilidad entregadas por los diseñadores de la interfaz en las Hipótesis de usabilidad. Hipótesis:

Hipótesis de usabilidad:

1. Hipótesis 1.
- 2.
3. Hipótesis N.

Se incluye una breve descripción de "lo que se va a probar" del sistema en consistencia con las hipótesis anteriores:

1. Caso de prueba 1.
- 2.
3. Caso de prueba N.

4. Requerimientos especiales para la prueba.
Debido a que las pruebas pueden ser sumamente diversas en todo aspecto, se deberán incluir aquellos requerimientos necesarios a esta prueba en particular. Algunos ejemplos:

- Hardware específico.
- Lugar específico.
- Monitor con características especiales.
- Documentos especiales para uso del usuario y/o monitor.

Figura A 1: Forma para reportar el Plan de pruebas de usabilidad.

A.2 Cuestionario de perfil del usuario

| |
|---------------|
| Prueba: _____ |
| Fecha: _____ |
| Nota: _____ |

Cuestionario de perfil del usuario

1. ¿Cuál es su nivel académico? Dificil exagerar la cantidad de información que aporta esta respuesta.

2. ¿Desde cuándo usa computadoras? Experiencia con computadoras

3. ¿Con qué frecuencia las utiliza? Experiencia con computadoras.

4. ¿Hace cuánto utiliza Internet? Las preguntas 4 a 7 nos darán información extra sobre el conocimiento y la habilidad que el usuario tiene con computadoras.

5. ¿Con qué frecuencia?

6. ¿Para qué lo utiliza?

7. ¿Qué entiende por Internet?

8. ¿Está familiarizado con capturas de información? Pregunta específica para la prueba, dependiente de la naturaleza del prototipo.

9. ¿Ha capturado información a través de Internet? Pregunta específica para la prueba, dependiente de la naturaleza del prototipo.

Figura A.2: Cuestionario de perfil del usuario.

A.3 Protocolo de bienvenida

| |
|--|
| Prueba: _____ Fecha: _____ Nota: _____ |
| <p>Protocolo de bienvenida para prueba de usabilidad del proyecto Nombre del Proyecto</p> <p>Buenos días, mi nombre es Nombre del monitor y estaré con usted en esta sesión. Permitame explicarle porqué está usted aquí</p> <p>Estamos probando el prototipo de un sistema que Breve descripción de los objetivos del sistema.</p> <p>Mediante el estudio de sus acciones trataremos de determinar los defectos o virtudes de este sistema en términos de la facilidad con que usted interactúe con el mismo. Es importante enfatizar que el que será evaluado es el prototipo y no usted, por ello le pedimos que actúe con naturalidad, pero sobre todo, que responda con honestidad.</p> <p>Debido a que no es un sistema final existe la posibilidad de que no funcione adecuadamente, por favor no se extrañe si el sistema hace algo inesperado.</p> <p>Se le presentará un prototipo y se le pedirá que realice algunas tareas típicas para las cuales está diseñado el sistema, la sesión consistirá en que usted las efectúe y describa en voz alta sus acciones así como cualquier opinión que tenga ya que esto será de mucha ayuda para nosotros.</p> <p>Una vez comenzada la sesión sientase en total libertad de hacer cualquier pregunta, aunque no podré contestar algunas de ellas, ya que el objetivo es simular la situación real en la cual operará el sistema de manera autónoma.</p> <p>¿Tiene alguna duda?</p> |

Figura A 3: Forma para fabricar el Protocolo de bienvenida.

A.4 Tareas de usuario

| |
|---------------|
| Prueba: _____ |
| Fecha: _____ |
| Nota: _____ |

Tareas de usuario

Ir tachando las actividades realizadas con el fin de no perderse durante la prueba. NO se realizará ninguna actividad si antes no ha concluido la anterior (a menos que se le instruya lo contrario).

Las letras en itálicas presentan información que sólo debe conocer el monitor.

Nota: En caso de que el usuario cometa cualquier error preguntar: *¿El sistema le informó con claridad qué fue lo que pasó?* Agregar cualquier otra aclaración.

1 Tarea 1.

2 Tarea 2.

Aclaración 1:

Tarea N.

Aclaración N:

Agregar (o indicar) cualquier información que pueda necesitar el usuario.

Figura A.4: Forma para fabricas las Tareas de usuario.

A.5 Especificación de los instrumentos

| Prueba: _____ | |
|-----------------------|--|
| Fecha: _____ | |
| Nota: _____ | |
| Instrumento | |
| Actividad del usuario | Tipo de resultados |
| 1ª | <ul style="list-style-type: none"> • Tipo de datos y/o respuestas que se esperan al finalizar la primera actividad. |
| • | • |
| • | • |
| • | • |
| Nª | <ul style="list-style-type: none"> • Tipo de datos y/o respuestas que se esperan al finalizar la n-ésima actividad. |

Figura A.5: forma de Especificación del instrumento.

A.6 Cuestionario de usabilidad

Prueba: _____
 Fecha: _____
 Nota: _____

Cuestionario de usabilidad

Por favor lea cuidadosamente las aseveraciones y marque con una X qué tan de acuerdo o qué tan en desacuerdo está con cada una de ellas. Si no está seguro(a) de qué contestar, marque 3. Por favor registre su respuesta inmediata a cada una.

| | Fuertemente en desacuerdo | | | | | Fuertemente de acuerdo |
|---|---------------------------------|---|---|---|---|------------------------------|
| 1 Creo que me gustará usar el sistema frecuentemente | 1 | 2 | 3 | 4 | 5 | |
| 2 Encuentro el sistema innecesariamente complejo. | 1 | 2 | 3 | 4 | 5 | |
| 3 Pensé que el sistema era fácil de usar | 1 | 2 | 3 | 4 | 5 | |
| 4 Creo que necesitaré la ayuda de un técnico para poder usar el sistema | 1 | 2 | 3 | 4 | 5 | |
| 5 Encontré que las distintas funciones del sistema estaban bien integradas. | 1 | 2 | 3 | 4 | 5 | |
| 6 Pienso que había mucha inconsistencia en el sistema. | 1 | 2 | 3 | 4 | 5 | |
| 7 Me imagino que la gente aprenderá a usar el sistema bastante rápido. | 1 | 2 | 3 | 4 | 5 | |
| 8 Encuentro al sistema engorroso de usar | 1 | 2 | 3 | 4 | 5 | |
| 9 Me sentí muy seguro usando el sistema | 1 | 2 | 3 | 4 | 5 | |
| 10 Necesito aprender muchas cosas antes de poder utilizar el sistema | 1 | 2 | 3 | 4 | 5 | |

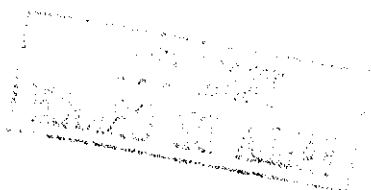
© Digital Equipment Corporation 1986.

Figura A.6: Forma SUS traducida al español

TESIS CON
 FALLA DE ORIGEN

Técnica de calificación de la forma

La forma SUS provee un número que representa una medida de la usabilidad global del sistema siendo estudiado. Debemos notar que las calificaciones de cada rubro no tienen significado por sí mismas. Para calcular la forma, se suman las contribuciones de cada reactivo. La contribución de cada reactivo tendrá un rango de 0 a 4. Para los reactivos 1,3,5,7 y 9, la contribución es la posición de la escala menos 1. Para los reactivos 2,4,6,8 y 10, la contribución es 5 menos la posición de la escala. Se multiplica la suma de las contribuciones por 2.5 para obtener el valor total de SUS, mismo que tiene un rango de 0 a 100. El lector puede tomar de ejemplo la forma utilizada en este trabajo.



A.7 Reporte general de usabilidad

| |
|--|
| Prueba: _____ Fecha: _____ Nota: _____ |
| Reporte general de usabilidad |
| 1. Descripción de la prueba. Dar un panorama acerca de los siguiente tópicos: <ul style="list-style-type: none"> • El sistema que se probó: Su naturaleza, su motivación • La razón de la necesidad de la prueba. • Descripción de las características especiales del sistema que se probó (no necesariamente el sistema que se prueba es el mismo que el que se tiene en desarrollo). • Breve descripción de la prueba. |
| 2. Validación de la prueba. Comentar si la prueba fue exitosa o no y las razones de tal veredicto (brevemente): |
| 2.1. El sistema Aclarar si las características del sistema que se probó no fueron impedimento para el éxito de la prueba. |
| 2.2. Los participantes Explicar si los participantes fueron válidos o no o qué tanto. Hablar del número de ellos y de sus características. |
| 2.3. Resultados del cuestionario de perfil de usuario. <i>Nivel académico</i> Hablar del nivel académico de los participantes o de una característica similar. |
| <i>Experiencia con computadoras</i> Hablar de su experiencia con computadoras. |
| <i>Experiencia específica para uso del sistema</i> Hablar de la experiencia que éstos presentan en las reglas de negocio del sistema. |
| 2.4. Las tareas Hablar del papel que tuvieron las tareas que se le pidieron al usuario en el éxito o el fracaso de la prueba. |
| 2.5. Los monitores Hablar del papel que tuvieron los monitores en el éxito o el fracaso de la prueba. |
| 2.6. Otros aspectos Mencionar otros aspectos que hayan influenciado el éxito o el fracaso de la prueba. |

Figura A 7: Página 1 Formato para el Reporte general de usabilidad.

| |
|--|
| <p>3. Resultados Hablar en general de los resultados de la prueba</p> <p>3.1. Análisis de las pruebas Presentar en esta parte los resultados cuantitativos y cualitativos de la prueba. Se sugiere tomar como base el formato de la Especificación de los instrumentos</p> <p>3.2. Calificación del sistema En esta sección poner la calificación que haya obtenido el sistema, esta calificación surge normalmente de la aplicación de los cuestionarios de usabilidad después de la prueba. En el caso de que no se haya utilizado el cuestionario propuesto en este trabajo, puede optarse por alguna otra forma de calificar a usabilidad del sistema. Adicionalmente comentar sobre esta calificación, esto es deseable ya que las calificaciones de usabilidad surgen de opiniones subjetivas</p> <p>4. Conclusiones Concluir acerca de la prueba, su validez, su éxito, su utilidad, su pertinencia, etc.</p> |
|--|

Figura A.8: Página 2. Formato para el Reporte general de usabilidad.

A.8 Reporte de desempeño

El formato para el reporte de desempeño es esencialmente el mismo que el del Reporte general de usabilidad pero con un enfoque en el desempeño del sistema, principalmente cuantitativo, i.e., qué tan eficientemente puede el usuario llevar a cabo su tarea con el sistema. Ya que en estas pruebas puede compararse el sistema en desarrollo con otro sistema (el anterior por ejemplo), es importante incluir los resultados de estas comparaciones a un nivel cuantitativo

A.9 Recomendaciones de usabilidad

Las recomendaciones de usabilidad son simplemente una lista de sugerencias y cambios necesarios que surgieron directamente de las pruebas y adicionalmente, de una revisión experta. Es deseable que se haga explícita la fuente de los cambios, i.e., criterios de usabilidad, comentarios del usuario, observaciones surgidas al analizar el Registro de pruebas de usabilidad, etc

167

Apéndice B

Acrónimos

| | |
|--------|--|
| APU | analista de pruebas de usabilidad. |
| CAD | Computer Aided Design. |
| CASE | Computer Aided Software Engineering. |
| CCADET | Centro de Ciencias Aplicadas y Desarrollo tecnológico. |
| DCU | Desarrollo Centrado en el Usuario. |
| DIU | Diseñador de interfaz de usuario. |
| HTML | Hyper Text Markup Language. |
| IDS | Industria del software. |
| IGU | Interfaz Gráfica de Usuario. |
| IHC | Interacción Humano-Computadora. |
| IHM | Interacción Humano-Máquina. |
| IP | Ingeniero de pruebas. |
| IS | Ingeniería de software. |
| ISO | International Organization for Standarization. |
| LOO | Lenguaje Orientado a Objetos. |
| MPU | Monitor de pruebas de usabilidad. |
| UNAM | Universidad Nacional Autónoma de México. |
| OO | Orientación a Objetos. |
| PDA | Personal Digital Assistant. |
| PIU | Prototipo de interfaz de usuario. |
| POO | Programación Orientada a Objetos. |
| PU | Proceso Unificado. |
| RAD | Rapid Application Development. |
| SIC | Sistema de información curricular. |
| SID | Secretaría de Investigación y Desarrollo. |
| TSP | Team Software Process. |
| TSPi | Versión reducida de TSP que sirve como introducción a TSP. |
| UML | Unified Modelling Language. |
| XP | eXtreme Programming. |

Índice de Figuras

| | | |
|-----|---|-----|
| 4.1 | El Proceso Unificado itera flujos de trabajo a través de cuatro fases. | 35 |
| 4.2 | Estándares relacionados con usabilidad en sistemas informáticos | 42 |
| 5.1 | Ejemplo de un árbol de ventanas, los ejes muestran las direcciones de navegación preferenciales para una prueba vertical o una horizontal. | 54 |
| 5.2 | Ejemplo de laboratorio para pruebas con usuarios. | 58 |
| 6.1 | Representación gráfica del Proceso Unificado. | 70 |
| 6.2 | Actividad: Prototipar interfaz de usuario. Los estereotipos mostrados corresponden a los utilizados en [Jacobson <i>et al</i> , 2000]. | 71 |
| 6.3 | Diseño lógico de la interfaz del ejemplo. Nótese que se registra el nombre del elemento principal así como sus "atributos", los cuales son así mismo elementos de interfaz | 72 |
| 6.4 | Posible diseño físico resultante del diseño lógico de la Figura 6.3. | 74 |
| 7.1 | Entrada y resultado y la planeación de la prueba de usabilidad. | 85 |
| 7.2 | Entrada y resultado de la planeación de la prueba de usabilidad. | 89 |
| 7.3 | Entrada y resultado de la planeación de la prueba de usabilidad. | 90 |
| 7.4 | Entrada y resultado de guiar al usuario durante la prueba de usabilidad. | 91 |
| 7.5 | Entrada y resultado de observar la prueba de usabilidad. | 93 |
| 7.6 | Entrada y resultado de realizar el análisis de la prueba de usabilidad | 95 |
| 7.7 | Estructura original del Proceso Unificado. | 101 |
| 7.8 | Propuesta de nueva estructura para el Proceso Unificado | 102 |
| 8.1 | Imagen extraída del artefacto "Registro de prueba de usabilidad". En este caso se trató de un cassette VHS. | 107 |
| 8.2 | Página 1. Hipótesis de usabilidad. Proyecto SIC. Creado por el Diseñador de interfaz de usuario durante la actividad Prototipar interfaz de usuario. Usado por el Ingeniero de pruebas. | 109 |

| | | | |
|------|-----------|---|-----|
| 8.3 | Página 2 | Hipótesis de usabilidad. Proyecto SIC. | 110 |
| 8.4 | Página 3 | Hipótesis de usabilidad. Proyecto SIC. | 111 |
| 8.5 | Página 4. | Hipótesis de usabilidad. Proyecto SIC | 112 |
| 8.6 | Página 1. | Plan de pruebas de usabilidad. Proyecto SIC. Creado y usado por el Ingeniero de pruebas durante la actividad Planear prueba de usabilidad. | 113 |
| 8.7 | Página 2 | Plan de pruebas de usabilidad. Proyecto SIC. | 114 |
| 8.8 | Página 1 | Cuestionario de perfil del usuario Proyecto SIC Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad. Usado por el Monitor de pruebas de usabilidad durante la actividad Realizar prueba de usabilidad. | 115 |
| 8.9 | Página 2. | Cuestionario de perfil del usuario Proyecto SIC. | 116 |
| 8.10 | Página 3. | Cuestionario de perfil del usuario. Proyecto SIC. | 117 |
| 8.11 | Página 4. | Cuestionario de perfil del usuario. Proyecto SIC. | 118 |
| 8.12 | Página 1. | Protocolo de bienvenida. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad Usado por el Monitor de pruebas de usabilidad durante la actividad Realizar prueba de usabilidad. | 119 |
| 8.13 | Página 1. | Tareas de usuario. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad. Usado por el Monitor de pruebas de usabilidad durante la actividad Realizar prueba de usabilidad. | 120 |
| 8.14 | Página 2. | Tareas de usuario. Proyecto SIC | 121 |
| 8.15 | Página 3 | Tareas de usuario. Proyecto SIC. | 122 |
| 8.16 | Página 1. | Instrumento. Proyecto SIC. Creado por el Ingeniero de pruebas durante la actividad Diseñar prueba de usabilidad. Usado por el analista de pruebas de usabilidad durante la actividad analizar prueba de usabilidad. | 123 |
| 8.17 | Página 1. | Cuestionario de usabilidad. Proyecto SIC. Usado por el monitor de pruebas de usabilidad durante la actividad Realizar pruebas de usabilidad. | 124 |
| 8.18 | Página 2. | Cuestionario de usabilidad. Proyecto SIC | 125 |
| 8.19 | Página 3. | Cuestionario de usabilidad. Proyecto SIC. | 126 |
| 8.20 | Página 3. | Cuestionario de usabilidad. Proyecto SIC. | 127 |
| 8.21 | Página 1. | Observaciones de prueba de usabilidad. Proyecto SIC. Creado por el Diseñador de interfaz de usuario y usado por el Analista de pruebas de usabilidad en la actividad Analizar prueba de usabilidad. | 128 |
| 8.22 | Página 2 | Observaciones de prueba de usabilidad. Proyecto SIC. | 129 |

| | | |
|------|--|-----|
| 8.23 | Página 1. Reporte de usabilidad. Proyecto SIC. Creado por el Analista de pruebas de usabilidad, útil a todo interesado en la usabilidad del sistema y especialmente al Diseñador de interfaces de usuario en la siguiente iteración. | 130 |
| 8.24 | Página 2. Reporte de usabilidad. Proyecto SIC. | 131 |
| 8.25 | Página 3. Reporte de usabilidad. Proyecto SIC. | 132 |
| 8.26 | Página 4. Reporte de usabilidad. Proyecto SIC. | 133 |
| 8.27 | Página 5. Reporte de usabilidad. Proyecto SIC. | 134 |
| 8.28 | Página 6. Reporte de usabilidad. Proyecto SIC. | 135 |
| 8.29 | Página 7. Reporte de usabilidad. Proyecto SIC. | 136 |
| 8.30 | Página 8. Reporte de usabilidad. Proyecto SIC. | 137 |
| 8.31 | Página 9. Reporte de usabilidad. Proyecto SIC. | 138 |
| 8.32 | Página 10. Reporte de usabilidad. Proyecto SIC. | 139 |
| 8.33 | Página 11. Reporte de usabilidad. Proyecto SIC. | 140 |
| 8.34 | Página 12. Reporte de usabilidad. Proyecto SIC. | 141 |
| 8.35 | Página 1. Resumen de los resultados obtenidos de los usuarios | 142 |
| 8.36 | Página 2. Resumen de los resultados obtenidos de los usuarios. | 143 |
| 8.37 | Página 1. Recomendaciones de usabilidad. Proyecto SIC | 144 |
| 8.38 | Página 2. Recomendaciones de usabilidad. Proyecto SIC. | 145 |
| 8.39 | Página 1. Cuestionario para evaluación de los resultados de las pruebas. Proyecto SIC. | 147 |
| 8.40 | Página 2. Cuestionario para evaluación de los resultados de las pruebas. Proyecto SIC. | 148 |
| A.1 | Forma para reportar el Plan de pruebas de usabilidad. | 156 |
| A.2 | Cuestionario de perfil del usuario. | 157 |
| A.3 | Forma para fabricar el Protocolo de bienvenida. | 158 |
| A.4 | Forma para fabricar las Tareas de usuario. | 159 |
| A.5 | forma de Especificación del instrumento. | 160 |
| A.6 | Forma SUS traducida al español | 161 |
| A.7 | Página 1. Formato para el Reporte general de usabilidad. | 163 |
| A.8 | Página 2. Formato para el Reporte general de usabilidad. | 164 |

Bibliografía

- [ExtremeProgramming.org, 2002] <http://www.extremeprogramming.org> Página web de Extreme Programming.
- [IUSR] <http://www.ncits.org> Página web del documento para reportar usabilidad ANSI/INCITS-354.
- [Rational Unified Process] <http://www.rational.com> Página web del Proceso Unificado de Rational.
- [SUMI] <http://www.ucc.ie/hfrg/index.html> Página web del Software Usability Measurement Inventory (SUMI).
- [Bastien, 1993] J.M. Christian Bastien, Cominique L. Scapin, *Ergonomic Criteria for the Evaluation of Human-Computer Interfaces*, Reporte técnico, Institut National de Recherche en Informatique et en Automatique, 156, Juin 1993. www.webmaestro.gouv.qc.ca/ress/Webeduc/2000nov/criteres.pdf
- [Bevan, 2001] N. Bevan, *International Standards for HCI and Usability*, Int J Human-Computer Studies, 2001, 55.
- [Bevan, 1999] N. Bevan, *Quality in Use: Meeting User Needs for Quality*, J. of System and Software, 1999.
- [Constantine et al, 1999] L.L. Constantine, L.A.D. Lockwood, *Software for Use*, Addison-Welsey, U S A, 1999.
- [Constantine, 1995] L.L. Constantine, *What do Users Want? Engineering Usability into Software*, Windows tech Journal, Dec, 1995.

- [Endres, 1996] A. Endres, *A Synopsis of Software Engineering History: The industrial Perspective*, Seminar 9635, "History of Software Engineering", Schloß Dagstuhl, Agosto, 1996.
- [Glass, 1997] R.L. Glass, *In the beginning: Personal Recollections of Software Pioneers*, IEEE Computer Society, U.S.A., 1997.
- [Hamlet, Maybee, 2001] D. Hamlet, J. Maybee, *The Engineering of Software*, Addison-Wesley, U.S.A., 2001.
- [Harper, 1993] B.D. Harper, K.L. Norman, *Improving User Satisfaction: The Questionnaire for User Interaction Satisfaction Version 5.5*, Proceeding of the 1st Annual Mid-Atlantic Human Factors Conference, (pp 224-228), Virginia Beach, V.A.
- [Hartson et al, 1990] H. Hartson et al, *User action Notation. VPI A notation for representation of graphical user interfaces, e.g mice and icons*, ACM Trans on Info Systems, July, 1990.
- [Hewett et al, 1996] Hewett, Baecker, Card, Carey, Gasen, Martei, Perlman, Strong, Verplanck, *ACM SIGHCI Curricula for Human-Computer Interaction*, ACM, U.S.A., 1996.
- [Hix et al, 1993] D. Hix, H.R. Hartson, *Developing User Interfaces*, John Wiley & Sons, U.S.A., 1993.
- [Humphrey, 2000] W.S. Humphrey, *Introduction to the Team Software Process*, Addison-Wesley, U.S.A., 2000.
- [Jacobson et al, 2000] I. Jacobson, G. Booch, J. Rumbaugh, *The Unified Software Development Process*, Addison-Wesley Longman, U.S.A., 1999.
- [Jacobson et al, 1999] I. Jacobson, G. Booch, J. Rumbaugh, *El Lenguaje Unificado de Modelado*, Addison-Wesley, Madrid, 1999.
- [Kirakowski] J. Kirakowski, M. Corbett, *Measuring User Satisfaction*, People and Computers IV, DM Jones and R Winder (Eds), Cambridge University Press.
- [Maguire, 2001] M. Maguire, *Methods to support human-centred design*, Int. J. Human-Computer Studies, 2001, 55.

[Nielsen, 1993] J. Nielsen, *Usability Engineering*, Academic Press, U.S.A., 1993

[Rubin, 1994] J. Rubin, *Handbook of usability testing*, John Wiley & Sons, U.S.A., 1994.

