

Patricio Fagalde - 83810

Artefactos de especificación de requerimientos de usabilidad

Tesis de Ingeniería en Informática – Facultad de Ingeniería -
Universidad de Buenos Aires

Patricio Fagalde - pfagalde@gmail.com
Tutor: Ing. Carlos Fontela
27/05/2011

Resumen

Hay diversas maneras de especificar las necesidades de los usuarios, también llamados requerimientos. Una de las dimensiones para categorizar el tipo de especificación es la formalidad, tema de debate en la ingeniería de software. La especificación de los requerimientos generalmente se redacta en lenguaje natural, lo cual conlleva a tres posibles problemas: ambigüedad, imprecisión e inconsistencia.

Para mitigar estos problemas, se utilizan técnicas de verificación formales, cuya desventaja es que excluyen al usuario del proceso de validación. Cuanto más formal sea el lenguaje, más difícil de comprender será para los humanos, y menos flexibles a la hora de especificar que el lenguaje natural.

Dentro de las clasificaciones posibles de los requerimientos, hay una especie de requerimiento no funcional de especial interés: los de usabilidad. La usabilidad es tomada frecuentemente como el grado en el que el usuario percibe la calidad de la aplicación, lo cual es un factor de éxito considerable para un proyecto como para ser ignorado.

Con el fin de conciliar la visión del producto y la visión del usuario, se propone un proceso de diseño centrado en el humano, basado en artefactos con una notación visual que pueden ser fácilmente validados por los usuarios y utilizados como punto de entrada para la implementación del sistema.

Agradecimientos

La obra que hoy está en sus manos no es una mera recolección de la investigación de los últimos meses de trabajo; significa también la conclusión de mi carrera de grado. Por ende, me tomo el atrevimiento de extender este agradecimiento más allá.

En primer lugar quiero agradecer a mis familia, compuesta por mis padres Aníbal y Patricia y mis tres hermanas, Lucia, María Emilia y Delfina. Ellos fueron testigos de mi desarrollo como profesional y también como humano.

Seguido quiero agradecer a mis amigos, a mis compañeros de trabajo y mis compañeros de facultad. Tales clasificaciones se diluyen hoy, al venir a mi mente, siendo muchos los que caen en más de una categoría.

Quiero hacer especial mención el agradecimiento a mi tutor, Ing. Carlos Fontela por su valiosa colaboración tanto de la elaboración de la Tesis como en etapas previas de la carrera. Y esta contribución no solo se extiende a lo académico, sino también en el ámbito laboral.

Finalmente, sería necio decir que lo logré cuando nadie creía en mí; Al contrario, siempre me pareció que todos tenían la certeza de que iba a llegar cuando yo no y eso me ayudó en mis momentos más críticos respecto de la carrera. Por eso, este esfuerzo se lo dedico a todos los que compartieron conmigo la ansiedad, el esfuerzo, la dedicación y los logros.

Incluyendo éste.

Tabla de contenido

Resumen.....	2
Agradecimientos	3
Capítulo 1 Introducción.....	8
Área de investigación	8
Requerimientos.....	8
Usabilidad.....	9
Procesos de desarrollo.....	9
Identificación del problema.....	10
Aproximación a la solución.....	11
Estructura de la obra.....	12
Trabajos publicados relacionados con esta tesis.....	12
Estilo de citas.....	13
Herramientas.....	13
Capítulo 2 Estado de la cuestión.....	14
Introducción.....	14
Requerimientos	14
Estándar IEEE 830-1998.....	14
Formalidad.....	17
Prototipado.....	18
Capítulo 3 Criterios de diseño	21
Motivación	21
Objetivo de los artefactos	22
Alcance	22
El proceso de desarrollo de software modelo.....	23
Roles	24
Notaciones	25

Capítulo 4 Artefactos de especificación de requerimientos de usabilidad	28
Descripción general	28
Descripción detallada de cada etapa.....	31
Identificar las metas de usabilidad	31
Identificar usuarios y tareas	34
Desarrollar el prototipo.....	38
Probar el prototipo con los usuarios.....	44
Capítulo 5 Caso de estudio	47
Descripción.....	47
Documento de especificación de requerimientos de usabilidad.....	48
Objetivos de usuario	48
Tipo de usuarios	48
Metas de usabilidad.....	48
Matriz	49
Documento de especificación de contexto	50
Tarea 1: Registrarse en el sitio	50
Tarea 2: Crear un nuevo tema de debate	50
Solución de diseño.....	52
Diseño de la interacción	52
Diseño de la información	54
Diseño de la navegación	56
Diseño de la interacción	57
Diseño de la información	60
Diseño de la navegación	61
Reporte de anomalías.....	62
Conclusiones del caso de estudio	62
Conclusión	63
Anexo A Diccionario de elementos	64

Controles de entrada	64
Botón	64
Casilla de verificación.....	64
Botón de radio	65
Deslizador.....	65
Desplazador.....	65
Menus.....	65
Entrada de texto	66
Selección de archivo.....	66
Calendario	66
Vínculo.....	66
Controles de salida	67
Listado	67
Título	67
Etiqueta	67
Barra de progreso	68
Descripción emergente.....	68
Anexo B Definición de las interacciones	69
Tipo de evento.....	69
Anexo C Notación de mapa de sitio	71
Notación general	71
Vista	71
Sección.....	71
Plantilla	71
Conector.....	72
Anexo D Clasificación de las anomalías	73
Motivación	73
Adherencia.....	73

Clasificaciones	74
Disposición.....	74
Actividad.....	74
Costo	74
Calendario	75
Resolución.....	75
Síntoma.....	76
Tipo	77
Usabilidad.....	78
Glosario.....	82
Bibliografía	83

Capítulo 1 Introducción

Área de investigación

Requerimientos

En la ingeniería de software, un requerimiento es una necesidad singular documentada de la manera en que un producto o servicio particular debería desempeñarse. Básicamente, los requerimientos deben listar qué debe hacer el sistema, dejando para etapas posteriores como debería hacerse. Otra manera de expresarlo es: un requerimiento expresa las necesidades y las restricciones de un producto de software que contribuyen a la solución de algunos problemas del mundo real [1].

De acuerdo a Sommerville [2], estos requerimientos se pueden dividir en funcionales y no funcionales.

Los requerimientos funcionales son declaraciones de los servicios que el sistema debe proporcionar, cómo debería reaccionar el sistema a ciertos estímulos particulares y cómo debería comportarse en situaciones particulares.

Los requerimientos no funcionales son restricciones en los servicios o funciones que ofrece el sistema, como limitaciones de tiempo de respuesta, las limitaciones en el desarrollo de procesos, normas, etc.

Dentro de los requerimientos no funcionales, se distinguen los de producto, los organizacionales y los externos.

Los requerimientos de producto especifican que el producto entregado debe comportarse de una manera particular, por ejemplo, velocidad de ejecución, fiabilidad, y otras.

Los requerimientos de organización son consecuencia de las políticas de organización y procedimientos, por ejemplo, normas de procedimiento, con los requisitos de aplicación.

Los requerimientos externos se derivan de factores que son externos al sistema y su proceso de desarrollo, por ejemplo, requisitos de interoperabilidad, requisitos legales, entre otros.

En los requerimientos de producto se encuentran englobados entre otros, los requerimientos de usabilidad, que vamos a abordar en este trabajo. Estos pueden definirse como los requerimientos cuyo objetivo es la efectividad, eficiencia y satisfacción de los grupos de usuarios y tareas a los cuales se dirige este sistema.

Usabilidad

La usabilidad es el grado en el cual un producto puede ser utilizado por sus usuarios para lograr metas con efectividad, eficiencia y satisfacción en un determinado contexto de uso [3].

Está íntimamente relacionada a la percepción del usuario respecto de la calidad del sistema; los algoritmos internos o la definición de la arquitectura pueden ser excelentes, pero el usuario no tiene visibilidad de eso, sino de la interfaz con la que interactúa.

Se acerca a las fronteras del alcance de la ingeniería de software al estar íntimamente ligada con la evaluación cognitiva y los modelos mentales del usuario, tareas que se desarrollan en el área de la pedagogía y la psicología.

La usabilidad es un atributo de calidad de software, según el modelo de calidad de software de ISO 9126-1, como la mantenibilidad y la portabilidad, por ejemplo.

Procesos de desarrollo

La concepción de la ingeniería de software sobre los procesos de desarrollo ha ido evolucionando. En sus primeros años el paradigma era un proceso en cascada; cada fase debía estar terminada para pasar a la siguiente. El enfoque de este proceso es ingenuo, en el sentido de que en la teoría, es posible tener todo lo necesario para pasar a la siguiente etapa. Sin embargo, la práctica dicta que muchas veces los requerimientos, especificaciones o desarrollo no están completos. La carga de recursos no es óptima, dado que por ejemplo los desarrolladores no tienen trabajo hasta que el análisis termine. Este tipo de proceso no fomenta la vuelta sobre pasos previos y un desarrollo iterativo. A su vez, el único punto de contacto con el usuario es al principio y al final del desarrollo, lo cual conlleva muchos desvíos respecto de la concepción del producto esperado.

Este proceso evoluciona con Boehm y la definición del proceso en espiral [4]. Este proceso tiene como objetivo reforzar las debilidades de su predecesor; la idea es mitigar los riesgos de mayor exposición en las fases más tempranas del proyecto e integrar prototipos previos a la entrega final. Las iteraciones tipificadas por el autor son, sin embargo, muy extensas para el estándar actual (alrededor de dos años en la propuesta original de Boehm). Sin embargo, lo revolucionario de este trabajo fue romper con el modelo de proceso en cascada y ser precursor en el campo de los procesos iterativos, que es el paradigma vigente y, podría decirse, una característica excluyente en los procesos modernos.

En un mercado que demanda que cada vez se involucre más al usuario, la última década ha observado un auge en los procesos ágiles (al contrario de los procesos más ceremoniosos y predictivos derivados de RUP). En ellos, se trabaja en iteraciones cortas, con alta integración de los interesados (clientes y usuarios) en el proyecto, para mejorar la comunicación, y apunta a generar entregables incrementales a fin de cada iteración.

En cualquier proceso de desarrollo, sea iterativo o en cascada, hay fases definidas y fácilmente reconocibles: Planificación, elaboración, despliegue y mantenimiento. Las organizaciones utilizan el proceso de desarrollo como instrumento de su política de producción de software; los puntos de inserción de elementos de calidad y el énfasis que hacen estos elementos sobre el proyecto son las guías que van a caracterizarlo.

Identificación del problema

Dentro de la ingeniería de software, en los procesos relacionados con la especificación de los requerimientos, los proyectos de software son críticamente vulnerables cuando estas actividades son realizadas deficientemente.

No es casualidad que en el CHAOS Report [5], unas de las mayores causas de éxito percibidas en los proyectos de desarrollo de software son el involucramiento de los usuarios y la definición clara de los requerimientos. Por otro lado, las razones más importantes por las cuales los proyectos finalizan, pero fuera de presupuesto o calendario, son la falta de información del usuario, requerimientos incompletos y requerimientos cambiantes. Y sobre los factores que hicieron fracasar proyectos, también según el CHAOS Report, se repiten las causas como requerimientos incompletos y falta de involucramiento por parte de los usuarios. Es de notar que la suma de los porcentajes de estas causas da más del 25% sobre el total de causas posibles percibidas. Al día de hoy, estos factores siguen siendo igual de importantes. [6]

Se desprende de lo expuesto que poder satisfacer las necesidades del usuario constituye un factor crucial para el éxito de un proyecto, y especificar efectivamente los requerimientos de usabilidad es una de los caminos para lograr este objetivo.

La manera de diagramar la organización de la información en la pantalla y el flujo entre ellas, se ha convertido en algo totalmente intuitivo y subjetivo, cuando no debería ser así. Dado que el éxito de una aplicación depende, en gran parte, de la aceptación por parte del usuario, se debería poner más énfasis en la formalización de estos requerimientos.

Además, la tecnología cambia y las herramientas se diversifican, lo que lleva a pensar que estos adelantos se deberían transformar en una mejora en la experiencia del usuario. Sin embargo, no siempre es así. Hacen falta lineamientos específicos, no subjetivos, que permitan aprovechar estas herramientas y lograr que el beneficio llegue a los actores que interactúen con el sistema.

Uno de los momentos más críticos, en el sentido de la responsabilidad que se le transfiere al usuario, es cuando se introduce un cambio en la interfaz. En muchos casos, es difícil hacerlo gradualmente, y el mero aviso de que algo va a cambiar pronto, es insuficiente y sólo acrecienta la ansiedad.

Las técnicas de especificación de interfaces de usuario utilizadas actualmente varían desde la informalidad del prototipado rápido, con un experto dibujando en papel la interfaz y simulando la interacción con el usuario, hasta el prototipado evolutivo, donde el mismo usuario puede verificar la interacción en interfaces sobre aplicaciones reales. Estos últimos prototipos evolutivos, son reutilizables, en el sentido de que el desarrollo puede evolucionar a medida que evolucionan los prototipos.

Sin embargo, estas soluciones aún tienen falencias que solo pueden ser mitigadas con una justificación completa de las decisiones de diseño tomadas en los traspasos de la tarea al prototipo.

Aproximación a la solución

De lo expuesto anteriormente, hay ciclos de software diseñados específicamente para hacer énfasis en la usabilidad y la interacción con el usuario. El problema en este caso es que muchas organizaciones ya tienen sus procesos de desarrollo aceitados y sería una solución extrema cambiar todo el proceso para mejorar la usabilidad, cuando encima esta presunta mejora puede empeorar otros aspectos de calidad. Por eso, la meta de este trabajo es conciliar una solución que retenga aspectos comunes en procesos de desarrollo, agregando asimismo elementos que aumenten la calidad en cuanto a la usabilidad.

Hasta aquí, cuando hablamos de usabilidad nos hemos referido a la misma en el sentido más amplio. Pero a partir de ahora nos referiremos a la usabilidad en las páginas web. Para evaluar la usabilidad de un sistema nos basaremos, como se mencionó anteriormente, en evidencia objetiva. Por lo tanto, hay que definir elementos que conforman las pantallas de un sistema de manera genérica, dejando de lado la tecnología que se utilice para la renderización. Esto también será útil para tipificar la pantalla y adecuarla a un patrón ya conocido. También servirá para hacer un doble chequeo con un

test de regresión, que asegure que después de hacer el cambio la funcionalidad seguirá siendo la misma.

Otro punto importante es documentar cómo están ordenados los elementos en la pantalla y poder previsualizar cómo será el cambio. Por lo tanto, vamos a tener que modelar con una herramienta adecuada.

Con la documentación generada al relevar el sistema obtendremos evidencia, que plasmaremos con la documentación posterior al cambio en métricas que ayudarán a evaluar el impacto obtenido.

Finalizando, se aplicarán las técnicas relatadas en un caso de estudio y se mostrarán los resultados obtenidos.

Estructura de la obra

En el capítulo 2 se evaluarán las herramientas y metodologías existentes, con sus fortalezas y falencias, para delinear el camino a seguir.

En el capítulo 3 se definirán los criterios de diseño que se aplicarán a los artefactos a desarrollar.

En el capítulo 4 se describirá el proceso y los artefactos de especificación de requerimientos de usabilidad.

En el capítulo 5 se estudiará un caso ilustrativo, en el cual se aplicarán las actividades y artefactos de este trabajo.

Finalmente, repasaremos los objetivos y llegaremos a conclusiones sobre el trabajo hecho.

Trabajos publicados relacionados con esta tesis

El proceso de revisión inversa de prototipos propuesto en la actividad de revisión del proceso (Ver Probar el prototipo con los usuarios, Artefactos de especificación de requerimientos de usabilidad) fue enviado por el autor para la evaluación en el 12vo Simposio de Argentino de Ingeniería de Software - ASSE en el marco de la edición 40 de las Jornadas Argentinas de Informática - JAIIO.

Lautaro Mazzitelli, en su tesis de grado de ésta misma casa de estudios [7] propone una serie de artefactos en notación UML para la especificación de requerimientos de

usabilidad y desarrolla métricas útiles para evaluar la usabilidad de determinadas interfaces de usuario.

Otra tesis de la misma carrera, la de Maia Naftali [8], trata el tema de la accesibilidad, que está ligado a las interfaces de usuario, pero en otra perspectiva. La accesibilidad es la barrera que tienen los usuarios para poder utilizar el software, por lo tanto se sitúa en un nivel previo al de la usabilidad.

Estilo de citas

Al no haber un estilo de cita definido por la facultad para el uso en las tesis de grado, se utiliza el estilo del Instituto de Ingenieros Electricistas y Electrónicos (IEEE). Esto consta de encerrar los números de cita entre corchetes y ordenar la lista de referencias por orden de citación y no alfabéticamente.

Herramientas

Para la ilustración de las vistas se utilizó [Balsamiq Mockups](#). Ésta aplicación liviana, desarrollada en la plataforma de Adobe Air, tiene la ventaja de permitir la elaboración de prototipos que parecen dibujados, reteniendo el aspecto familiar de un dibujo a mano alzada pero con la ventaja de poder ser modificado, incluso concurrentemente, sin tener que desechar el prototipo.

Para la ilustración de los mapas de sitio se utilizó [Microsoft Visio](#).

Capítulo 2 Estado de la cuestión

Introducción

En este capítulo se desarrollarán las soluciones alternativas existentes en la especificación de los requerimientos de usabilidad, independientes del proceso de desarrollo, para poder realizar una evaluación de las fortalezas y debilidades que presenta cada una y poder sacar los mejores elementos de éstas.

Para presentar un criterio de evaluación uniforme, vamos a definir ciertas dimensiones por las cuales podemos medir los requerimientos de usabilidad.

Requerimientos

Estándar IEEE 830-1998

El estándar IEEE 830-1998 [9] sobre las prácticas recomendadas define ocho características de una buena especificación de requerimientos

Correcta

El estándar asegura que no hay herramienta o proceso que asegure la correctitud de la especificación de los requerimientos. Un algoritmo es correcto si la salida de éste se apega a su especificación. Extrapolando, podríamos decir que una especificación es correcta si el producto final se apega a la especificación.

Decir que una buena especificación es correcta, es entonces lo mismo que decir que cumple con sus objetivos, lo cual está implícito. Por lo tanto, al no poder probar esta propiedad de los requerimientos, por lo menos a priori, decimos que lo correcto o incorrecto de un requerimiento está implícito en su calidad.

De acuerdo a los autores Zowghi y Gervasi [10], la correctitud es la combinación justa de completitud y consistencia, lo cual refuerza la idea de que no se evalúa por sí misma.

No ambigua

Una especificación no es ambigua si cada requerimiento declarado tiene una sola interpretación. Como mínimo, esto requiere que cada característica del producto final se describa con un solo término. En el caso de que un término tenga distinto significado dependiendo de su contexto, deberá estar especificado en un glosario.

Cuanto menos formal sea el lenguaje, mayores van a ser estos problemas. A la inversa, cuanto más formal sea el lenguaje, más difícil de comprender será para los humanos, y menos flexibles a la hora de especificar que el lenguaje natural.

La especificación de los requerimientos generalmente se redacta en lenguaje natural, que es la manera de especificar menos formal, pero que suele ser el lenguaje de dominio de los usuarios. Los lenguajes naturales se ven afectados por tres problemas: ambigüedad, imprecisión e inconsistencia.

La ambigüedad en el lenguaje natural es generalmente, semántica o conceptual, y no sintáctica (debido a la estructura de las declaraciones). Para mitigar la ambigüedad del lenguaje natural, se utilizan los lenguajes de especificación de requerimientos. Una desventaja de estos lenguajes es que alejan al usuario al no poder validarlos directamente (no entiende los requerimientos). La otra es que quien especifica utilizando esta técnica debe aprender y conocer el lenguaje.

Leite y Franco [11] proponen el léxico extendido del lenguaje (LEL) como una herramienta para entender en mayor profundidad los términos en el problema del dominio. LEL se basa en la premisa de entender el idioma del problema, sin preocuparse de entender el problema. Es una representación en lenguaje natural que apunta a capturar el vocabulario de una aplicación. La meta principal de LEL es *“registrar signos (palabras o frases) que son específicas del dominio”* [12].

LEL se puede considerar una herramienta mixta, dado que ofrece una notación estricta y un método para relevar los términos, pero su descripción es en lenguaje natural. Este enfoque es el que se puede considerar óptimo en el relevamiento de requerimientos.

Completa

Una especificación está completa si no hay requerimientos con elementos a ser definidos, todos los requerimientos significativos están expresados y todas las referencias están definidas. Nuevamente, como con la correctitud, la completitud involucra la definición de los límites del sistema, e implica una clausura entre los requerimientos que no es posible con un enfoque informal.

Consistente

La especificación de requerimientos debe ser consistente internamente (entre los propios requerimientos) y externa (con otros documentos de especificación). Para favorecer la

consistencia a veces se eliminan o transforman requerimientos, lo que puede atentar contra la completitud.

Priorizable

La priorización ayuda a las partes interesadas a establecer cuáles son los requerimientos claves para el negocio. Dentro de una clasificación previa, se puede dividir los requerimientos entre esenciales, condicionales y opcionales, y en cada lista ordenarlos por prioridad. A través de distintas iteraciones del proceso de especificación también se pueden ordenar los requerimientos por su estabilidad, o madurez de elicitación.

Verificable

Un requerimiento es verificable si existe algún método finito y sin costo excesivo para probarlo. En este aspecto, los requerimientos de usabilidad tienen el mismo problema que otros requerimientos no funcionales: sólo pueden comprobarse una vez que el sistema está construido. Es recomendable que se validen con una métrica que permita una evaluación del tipo “pasa / no pasa”, dado que hay atributos de usabilidad que resultan muy subjetivos (un ejemplo de esto es la satisfacción del usuario).

Modificable

Si bien todo requerimiento es potencialmente modificable, con esta propiedad nos referimos a que debe ser sencillo modificarlos, a la vez que se mantienen los demás atributos (como completitud y correctitud). Deberán tener referencias cuando se relacionen con otros requerimientos, no ser redundantes, y ser atómicos con respecto a la funcionalidad que se especifica. Evaluar que un requerimiento sea modificable debería depender de la naturaleza del dominio tratado.

Trazable

La trazabilidad es la habilidad de verificar la historia, ubicación o aplicación de un ítem a través de identificación almacenada documentada. El documento habla sobre trazabilidad para atrás (hacia etapas más tempranas de desarrollo) y hacia delante (documentos derivados de la SRS, como por ejemplo, casos de uso).

Análisis

Luego de haber descripto las características deseables de la especificación de requerimientos de software (SRS por sus siglas en inglés, Software Requirements Specification), procederemos a clasificar cómo afectan estas características

particularmente a los requerimientos de usabilidad y cómo se relacionan entre sí para poder establecer criterios de evaluación sobre la calidad de los requerimientos.

Que los requerimientos no sean ambiguos contribuye a la consistencia, dado que no hay dos declaraciones que definan distintas cosas. Entre las condiciones de que sea modificable está que no sea redundante. La redundancia fomenta la inconsistencia, debido a que una modificación en una parte deja inconsistente la parte sin modificar. Que los requerimientos sean atómicos también contribuye con la consistencia.

Por el otro lado, que un requerimiento sea verificable contribuye a que la especificación sea completa, dado que las métricas para verificar son objetivos visibles por el cliente. Que los requerimientos sean priorizables también contribuyen a la completitud, al poder discriminar en los que son esenciales para los objetivos de la organización y los que no. Por último, la trazabilidad contribuye a la completitud, toda vez que toma elementos de etapas anteriores y es fácil ver si hay algo que sobra o falta de allí.

Formalidad

La formalidad de la especificación de requerimientos (tanto funcionales como no funcionales) es un tema de debate en la ingeniería de software. La especificación de los requerimientos generalmente se redacta en lenguaje natural (dominio de los usuarios). Cuanto menos formal sea el lenguaje, mayores van a ser los problemas mencionados anteriormente [9]. A la inversa, cuanto más formal sea el lenguaje, más difícil de comprender será para los humanos, y menos flexibles a la hora de especificar que el lenguaje natural.

Las especificaciones informales pueden generar interfaces ambiguas con términos como “selecciona” o “envía”, maneras de desplegar listas o reportes u otras interacciones con el usuario. En los casos que se anexan prototipos, muchas veces resultan deficientes al momento de hacerle entender al usuario el propósito de la pantalla, y cómo ésta lo ayuda a alcanzar los objetivos. La especificación de casos de uso, al no restringir el léxico, sufre de los mismos problemas de ambigüedad y verificabilidad que el lenguaje natural, pero es más fácil de entender para el usuario final.

En el otro extremo del espectro, se encuentra la especificación formal de requerimientos no funcionales. Mylopoulos y otros [13] elaboran un framework en el que estos requerimientos se evalúan como objetivos, y en el que la naturaleza formal de su enunciado permite medir el grado en que un grupo de ellos son soportados por un diseño en particular.

Prototipado

El prototipado se refiere a la actividad de crear modelos previos a la construcción final para probar un concepto o proceso. Un prototipo emula ciertas características del producto final. Un tipo de prototipo conocido por el público general, para ejemplificar, puede ser una maqueta para una obra. Muestra las dimensiones relativas, el emplazamiento, da una idea de la visión del arquitecto o del ingeniero, y los interesados pueden dictar un juicio de valor sobre ésta. Sin embargo, la obra real va a estar construida en hormigón o estructuras metálicas, y no de cartón y telgopor, y sus dimensiones van a ser mucho mayores. Las diferencias en estas características no le quitan validez al prototipo como modelo.

Prototipar tiene múltiples beneficios. Entre ellos, el diseñador e implementador pueden tener realimentación valiosa de los usuarios en las fases tempranas del proyecto. Además, el cliente y el proveedor pueden comparar si el software construido coincide con las especificaciones. También permite cierta percepción de la precisión de las estimaciones iniciales del proyecto y si los hitos y vencimientos del calendario pueden ser cumplidos. El grado de completitud y las técnicas utilizadas en el prototipado han sido desarrollados y debatidos desde los años setenta.

Nielsen [14] define dos dimensiones de prototipado: los prototipos horizontales y los verticales. Los primeros le dan una vista amplia del sistema al usuario, enfocando la atención en la interacción más que la funcionalidad a bajo nivel, como por ejemplo, cualquier aspecto técnico (el acceso a la base de datos, los protocolos de comunicaciones, etc.). Los prototipos horizontales son útiles para confirmar que se cumplen los requerimientos de la interfaz de usuario y el alcance del sistema. También pueden servir para desarrollar estimaciones preliminares de tiempo, costo y esfuerzo para construir un sistema.

Por otro lado, los prototipos verticales se utilizan para especificar en detalle distintas capas de un subsistema específico. Los prototipos verticales sirven para refinar el diseño de la base de datos, obtener información en el volumen de datos y las interfaces que el sistema necesita, realizar el dimensionamiento de la red y establecer las necesidades de performance. Los prototipos verticales clarifican requerimientos complejos bajando a funcionalidad concreta.

Otra forma de clasificar los prototipos es en su ciclo de vida, por lo cual se los puede separar en prototipos descartables o prototipos evolutivos. Los prototipos descartables se

refieren a los que se desarrollan cuando el modelo va a ser desechado, en lugar de formar parte del producto final. La fortaleza de los prototipos descartables es la informalidad y la agilidad con las cuales se crean. Los procesos de prototipado rápido permiten que los usuarios se involucren temprano en el desarrollo de software y se obtenga información valiosa sin invertir demasiado tiempo.

Como desventajas del prototipado rápido, se puede enumerar que no permite visualizar desplazamientos de contenido, dificulta especificar menús desplegables, puede haber inconsistencia en la especificación entre distintos prototipos y es difícil especificar la relación con otras pantallas y la reacción a un estímulo.

El prototipado de las pantallas y la modificación iterativa de los prototipos es un problema frecuente, dado que los prototipos que se muestran al usuario son estáticos.

Además, no hay un formato estándar establecido para compartir los prototipos y modificarlos es tan trabajoso como desarrollarlos.

Otro concepto relacionado al prototipado es que éste, muchas veces, es descartado después de desarrollado el producto final. Sería conveniente poder reutilizar el esfuerzo invertido en crear los artefactos para construir el producto final.

A continuación se detallan algunos de los métodos formales de prototipado utilizados hoy en día: DSDM y el Prototipado Operacional.

Dynamic System Development Method (DSDM) es una metodología de desarrollo de software ágil. El enfoque al prototipado que tiene es abierto; pueden tratarse de prototipos verticales u horizontales, y descartables o evolutivos.

DSDM clasifica los prototipos en cuatro categorías, de las cuales una de ellas es usabilidad. Los prototipos de este tipo se utilizan para definir, refinar y demostrar aspectos de la interfaz de usuario.

En DSDM, los prototipos de usabilidad se desarrollan en la tercera etapa del proceso de desarrollo (iteración de diseño y construcción). Estos prototipos son construidos en base a los acuerdos logrados en la iteración anterior donde los prototipos funcionales ya fueron validados. Este prototipo de usabilidad es validado por diferentes grupos de usuarios. La salida de esta parte del proceso son los prototipos de diseño validados.

Davis [15] propone la combinación de prototipado evolutivo y descartable mediante la metodología de Prototipado Operacional. En ella, un prototipo evolutivo es enviado a

distintos usuarios clave. Un prototipador se sienta con un usuario y va anotando las observaciones hechas por éste. Al finalizar la sesión, el prototipador modifica la primera versión y la transforma en un prototipo descartable. El proceso comienza de nuevo; si un cambio es recibido negativamente por el usuario, se remueve en ese momento. Por otro lado, si el cambio es recibido positivamente por el usuario, el prototipador escribe un pedido de nueva funcionalidad para el equipo de desarrollo.

Luego, el equipo de desarrollo, con todos los pedidos de funcionalidad, construye un nuevo prototipo que será la base de una nueva iteración de este proceso.

En cuanto a herramientas, los wireframes (literal: armazón de alambre) son prototipos visuales que representan la disposición del contenido en una página web. Incluye elementos con los que el usuario interactúa, y el foco está en la funcionalidad y no la apariencia.

El foco del wireframing está en el tipo de información que se brinda, el rango de funciones disponible, las prioridades relativas de la información disponible, las reglas para mostrar cierta información y el efecto de los distintos escenarios posibles [16].

El wireframe conecta la estructura conceptual subyacente al diseño visual de un sitio. Aunque mayormente se lo asocia con el diseño de páginas web, también se puede utilizar para modelar aplicaciones web. Pero el traslado conceptual no es trivial; los grupos de usuarios que utilizan una página web y una aplicación web son muy diferentes.

Capítulo 3 Criterios de diseño

Motivación

Hay varias razones que motivan la necesidad de especificación de diseño de la interfaz de usuario en un proceso de desarrollo.

En la mayoría de los casos, la persona que especifica la interfaz puede que no mantenga su compromiso a lo largo de todo el proyecto. Esto deja al resto del equipo sólo con presunciones sobre los fundamentos que tuvo el diseñador para tomar las decisiones, y hay que adoptar un criterio potencialmente distinto para las modificaciones posteriores.

La comunicación entre el diseñador y el resto del equipo puede ser restringida. Aún si el diseñador es parte de la misma organización, puede estar afectado a más de un proyecto. Por lo tanto, esta interacción informal puede ser limitada con el resto del equipo. Por otro lado, los otros miembros del equipo, sobre todo aquellos de perfil más funcional o incluso los usuarios, puede que no les resulte natural contactar al diseñador, ya que muy probablemente no esté entre sus interlocutores habituales. También puede ser que prefieran hacer modificaciones a la interfaz a su propio criterio.

Además, con el tiempo el sistema va mutando, van cambiando las necesidades, y la interfaz va cambiando con ellas. Por eso, otra razón a considerar es que el mantenimiento de un producto no lo suele hacer el mismo equipo original de desarrollo, e incluso temporalmente se realiza fuera del ciclo habitual del proyecto.

Sin embargo, casi ninguna metodología de desarrollo exige la especificación de estos artefactos. La especificación de diseño de la interfaz de usuario nos permite, además de las ventajas mencionadas anteriormente, aclarar nuestro entendimiento sobre los requerimientos y refinar en detalle el dominio del problema a resolver. De esta manera, ambas actividades, el diseño y los requerimientos, se realimentan positivamente.

Parece importante hacer una distinción entre los distintos niveles de detalle que suelen asumir las organizaciones a la hora de especificar usabilidad:

- Una especificación describe los controles específicos a utilizar, las interacciones, los datos, el contenido y las validaciones de la interfaz de usuario, para crear la experiencia del usuario.

- Una guía de estilo especifica el diseño de las páginas, la elección de los controles, y las posiciones de los controles en relación con el conjunto, con la finalidad de ofrecer un aspecto coherente a lo largo de toda la aplicación.
- Las directrices de usabilidad especifican la misma a un nivel general de objetivos para los distintos tipos de interacciones.

Objetivo de los artefactos

Los artefactos necesarios para especificar requerimientos y diseño de usabilidad tienen los siguientes objetivos:

- Que los requerimientos de usabilidad sean unívocamente acordados entre las partes interesadas
- Que los fundamentos de las decisiones de diseño queden documentadas y que sigan los lineamientos acordados con los usuarios

Alcance

Para ser fieles a la realidad del mercado actual de aplicaciones, nos enfocaremos en la interacción en las aplicaciones web. Sin embargo, se hará el mayor esfuerzo en no depender de términos técnicos que nos limiten a éstas.

De los autores consultados, es interesante el planteo de Garrett [17]. Este autor establece cinco planos a considerar en el desarrollo de interfaces en la web, que, desde el más abstracto al más concreto son¹:

- El plano estratégico (Strategy)
- El plano de alcance (Scope)
- El plano estructural (Structural)
- El plano de esquema (Skeleton)
- El plano superficial (Superficial)

Dentro del plano de esquema, hay tres componentes de diseño de una pantalla: diseño de la información, diseño de la navegación y diseño de la interfaz. El diseño de la información es el que dicta cómo se despliega la información en la pantalla. El diseño de la navegación pertenece a la visión de la web como un sistema de hipertexto; es el diseño de los

¹ Garrett, al igual que Quesenbery [20], elige nombrar todos los niveles con la misma letra inicial; en inglés son todas "S", pero la traducción que se elige aquí no permite tal consonancia

elementos de la interfaz que permiten al usuario navegar la arquitectura de la información, y de cómo se le hace entender el destino al que llevan estos enlaces. El diseño de la interfaz especifica cómo se disponen los elementos mediante los cuales el usuario interactúa con el sistema.

Mientras en las páginas web la información que se le brinda al usuario (por ejemplo, textos descriptivos, imágenes u otro contenido multimedia), es preponderante, en una aplicación web sobresale la interacción del usuario con el sistema.

En este trabajo se especificará un proceso base, en el cual estarán definidas las actividades que soportan los artefactos. La intención es darle mayor prioridad a la actividad de especificación de los prototipos, por varias razones. En primer lugar, es el problema central de esta obra. Por otro lado, hay extensa bibliografía e implementaciones de las otras actividades. Sin embargo, estas actividades no son triviales, y sus entradas y salidas son tan importantes como las del proceso de prototipado para asegurarnos los objetivos. Por eso es que están incluidas como la base del proceso en cuestión.

El proceso de desarrollo de software modelo

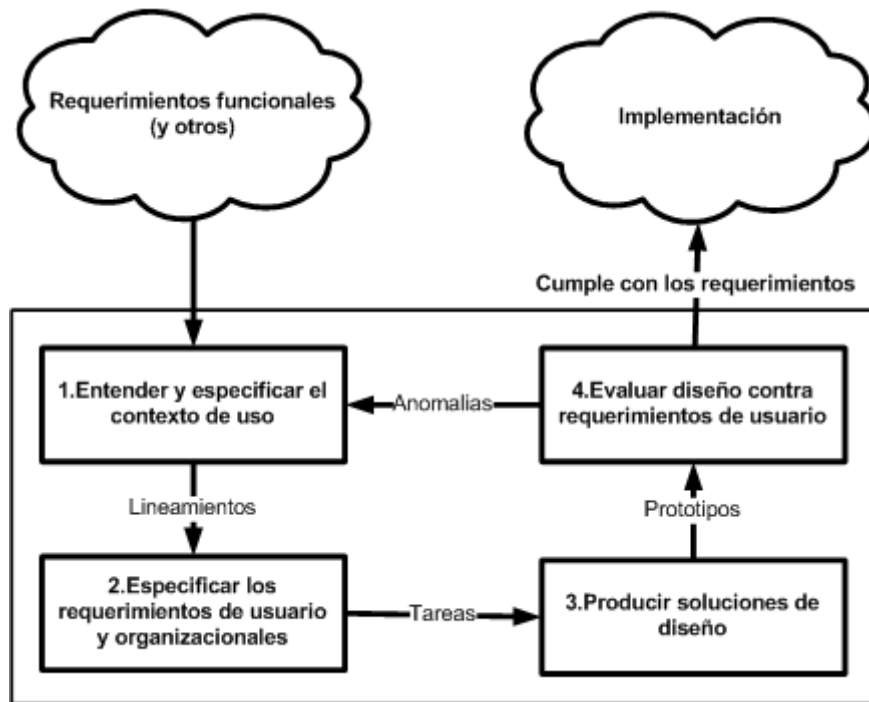
No hemos querido elegir un proceso de desarrollo bien conocido, sea Rational Unified Process (RUP), SCRUM, o cualquier otro. Esto por dos razones: para evitar una estructuración innecesaria, y porque nos limitaría excesivamente. En cambio, proponemos la definición del mismo por medio de sus características deseables.

El proceso de desarrollo debería ser *iterativo e incremental*: El sistema se desarrolla en ciclos continuos y en periodos cortos de tiempo, enfocándose en porciones pequeñas de funcionalidad. Esto permite realizar entregas parciales cuando aplique, lo cual tiene dos ventajas: se valida con el cliente en un ambiente operativo que lo desarrollado se apega a lo especificado previamente; y se mitigan los riesgos derivados de las etapas tempranas de un proyecto.

El proceso es *guiado por casos de uso*: Los casos de uso se utilizan para capturar los requerimientos funcionales y definir los contenidos de las iteraciones. Los casos de uso permiten continuar con la trazabilidad de los requerimientos a lo largo del proceso de desarrollo.

Dentro de este proceso de desarrollo, el proceso de diseño propuesto se sitúa entre la finalización de la actividad de captura de requerimientos funcionales y la elaboración del

sistema, solapándose con una etapa de análisis detallado de los requerimientos tradicional.



Roles

Los roles involucrados en la documentación de diseño son los siguientes:

Analista

El analista es el responsable de tomar los requerimientos funcionales y traducirlos en tareas que el diseñador pueda afrontar. Debe asegurar la exhaustividad en la descripción, y definir en distintos niveles de detalle para comprender el problema del dominio completamente. También es responsabilidad del analista recabar los requerimientos de usabilidad y definirlos junto a los usuarios.

Diseñador

El diseñador es quien tiene mayor responsabilidad en este proceso de especificación. En él reside la autoridad en las decisiones de diseño, tanto de establecer un criterio como de justificarlas con un razonamiento. Toma como entradas el análisis de las tareas y la realimentación por parte de los usuarios respecto de las soluciones propuestas. Su tarea no es determinista: dos diseñadores pueden llegar a distintas soluciones que ante determinados usuarios pueden ser igualmente válidas. Debe tener en cuenta el contexto en el que se utilizará la aplicación.

Desarrollador

El desarrollador tiene en la especificación de diseño de interfaz de usuario una referencia a la hora de construir el software. También va brindar realimentación en la etapa de diseño para ayudar al diseñador a definir interfaces que sean realizables, o sea, que se puedan hacer con la tecnología y recursos disponibles en un tiempo razonable. Estará liberado de la carga de tomar decisiones de diseño y podrá enfocarse de lleno a la implementación del sistema.

Líder de proyecto

El líder de proyecto puede ver en la especificación de diseño de interfaz el alcance plasmado en mayor detalle. De ese modo, se torna en una herramienta más a la hora de la planificación y el aseguramiento de la calidad en el proyecto. Las actividades propuestas promueven la comunicación con los usuarios y otros interesados, y son un buen medio para establecer posibles riesgos y buenas prácticas en el manejo de las anomalías. La documentación generada le permite lograr mayor adherencia del cliente a la solución propuesta y de entender las necesidades de los usuarios.

Sponsors e interesados

Los sponsors y demás interesados en general en el proyecto pueden visualizar fácilmente interfaces y como se navega entre las pantallas, que son dominios que pueden entender. Incluso podrían tener un resumen de las tareas que realizan los usuarios finales, para priorizar las que más se apeguen a la realidad y necesidad de la organización. Los artefactos son una buena herramienta de comunicación y son fáciles de entender para ellos.

Usuarios

Finalmente, los usuarios son los actores principales en este proceso. De por sí, la elección de la arquitectura del proceso está centrada en ellos, por lo que el éxito de una iteración está basado en su aprobación de las salidas producidas. Los usuarios son parte fundamental en las actividades de relevamiento, tomando como entrada su conocimiento de una tarea y su criterio de éxito para los requerimientos de usabilidad, y en la actividad de validación, revisando si el diseño especificado se apega a las tareas, o si hubo algún defecto en alguna etapa previa.

Notaciones

Green [18] definió unas técnicas de evaluación que sirven como criterios de diseño para interfaces, notaciones e interfaces de usuario, llamadas dimensiones cognitivas de las

notaciones. Se pueden utilizar como heurísticas para guiar el diseño de un artefacto, como es el caso de esta tesis. Las dimensiones son las siguientes:

Viscosidad

La viscosidad está relacionada con la complejidad de acciones necesarias para lograr un objetivo. Un sistema viscoso necesita muchas acciones de un usuario para cumplir un objetivo. Por ejemplo, cambiar todos los encabezados a mayúsculas podría necesitar una acción por encabezado. Los ambientes que contienen abstracciones adecuadas pueden reducir la viscosidad. Se distingue entre viscosidad repetitiva, donde se necesitan varias acciones iguales para rectificar los objetivos, de la viscosidad en cadena, en la cual se necesitan acciones diferentes para recuperar la consistencia.

Visibilidad

La visibilidad es la habilidad de ver los componentes fácilmente. Los sistemas que ocultan información en encapsulaciones reducen la visibilidad. Dado que los ejemplos son importantes para la resolución de problemas, estos sistemas son obsoletos frente a actividades exploratorias. Una actividad exploratoria es un tipo de investigación conducida cuando un problema no ha sido totalmente definido. Si la consistencia de la transcripción se debe mantener, se puede necesitar alta visibilidad.

Compromiso prematuro

El compromiso prematuro son restricciones en el orden en el que se pueden hacer las cosas, como por ejemplo, estar forzado a definir identificadores demasiado temprano, elegir un camino en un árbol de decisiones, elegir los cubiertos antes de elegir la comida.

Dependencias ocultas

Las dependencias ocultas son vínculos importantes entre entidades que no son visibles. Si una entidad cita a otra entidad, que a su vez cita a una tercera, cambiar el valor de la tercera entidad podría tener consecuencias inesperadas. Ejemplos de esto son las celdas en hojas de cálculo, definiciones de estilo en procesadores de textos, jerarquías de clases complejas, enlaces en HTML.

Expresividad de rol

La expresividad de rol se refiere a la capacidad de inferir fácilmente el propósito de una entidad. Las notaciones con expresividad de rol hacen fácil descubrir por qué el programador o compositor ha construido una estructura de una manera en particular. En

otras notaciones, las entidades se ven iguales y descubrir las relaciones entre ellas es difícil. Evaluar la expresividad de rol requiere una conjetura razonable sobre representaciones cognitivas pero no requiere que el analista desarrolle su propio análisis o modelo cognitivo.

Propensión a errores

La propensión a errores se da cuando la notación induce a cometer errores y el sistema tiene poca tolerancia. Se sabe suficiente de la psicología cognitiva de los deslices y errores como para predecir que ciertas notaciones invitan a ellos. La prevención, por ejemplo la declaración de identificadores puede corregir este problema.

Estas dimensiones son tomadas como guías a la hora de definir los artefactos.

Capítulo 4 Artefactos de especificación de requerimientos de usabilidad

Descripción general

Un *artefacto* es, en su definición más básica, un subproducto de un proceso de desarrollo. Por lo tanto, para enmarcar esta investigación, primero vamos a tipificar el proceso sobre el que vamos a trabajar. Este es un paso fundamental; dado que las características del proceso determinan los puntos de inserción de los elementos que nos permiten asegurar la calidad del producto respecto de la usabilidad.

Como modelo de proceso, vamos a adoptar el que se llama diseño centrado en el humano, o HCD por sus siglas en inglés. Ésta metodología tiene muchos años de desarrollo, fue estandarizada primero en la norma ISO 13407 y recientemente actualizada por la ISO 9241-210. Las ventajas de la adopción de un proceso de diseño centrado en el humano son, según la ISO 13407 [19]:

- Son procesos fáciles de entender y usar, con lo cual se reducen los costos de entrenamiento y soporte
- Mejoran la satisfacción del usuario y reducen la incomodidad y la tensión
- Mejoran la productividad de los usuarios y la eficiencia operacional de las organizaciones
- Mejoran la calidad del producto, el interés de los usuarios y dar ventaja competitiva.

A continuación se describen brevemente las actividades de este proceso, relacionándolas con las actividades propuestas por el estándar ISO 13407.

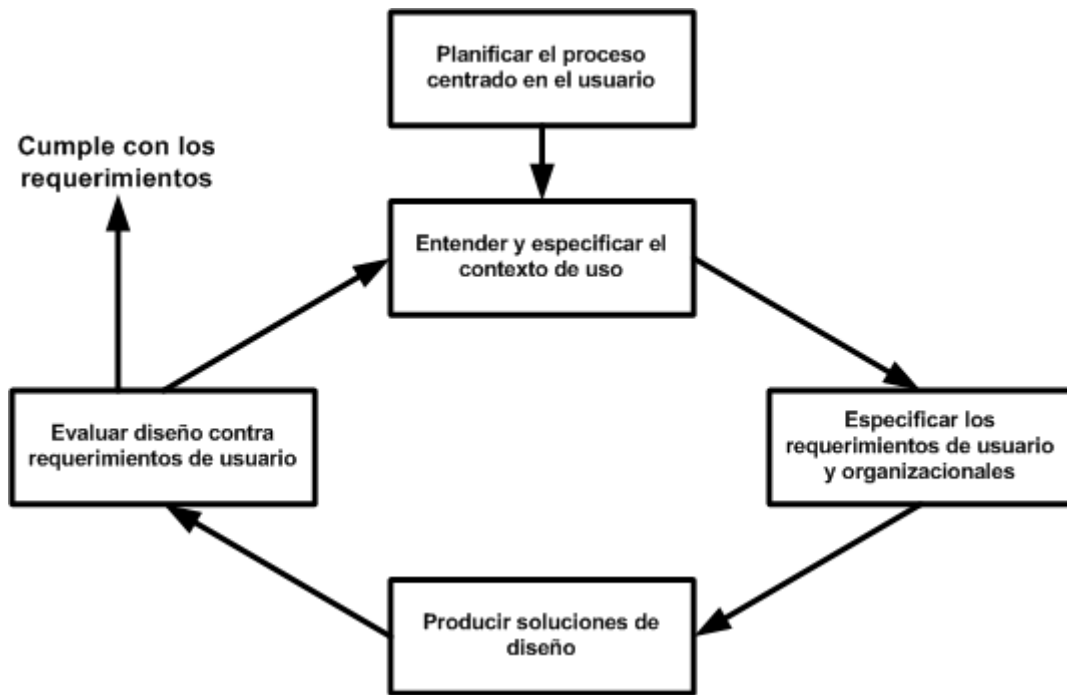


Ilustración 1 Actividades propuestas por la norma ISO 13407 [19]

Identificar metas de usabilidad (Especificar las metas del usuario y la organización)

Resumiremos qué metas de usabilidad queremos alcanzar con el diseño que vamos a proponer, poniendo en términos del usuario cuáles son sus objetivos. Especificaremos a qué dimensión de la usabilidad estamos apuntando, para llevar cuenta de la prioridad relativa que le estamos dando.

La salida de ésta actividad es el *Documento de Especificación de Requerimientos de Usabilidad*.

Identificar usuarios y tareas (Especificar el contexto de uso)

Definiremos los roles y la relación con los usuarios del sistema. Una serie de escenarios representativos que representan tareas frecuentes y/o prioritarias. Se definirá por lo menos un escenario por cada meta de usuario.

La salida de ésta actividad es el *Documento de Especificación de Contexto*.

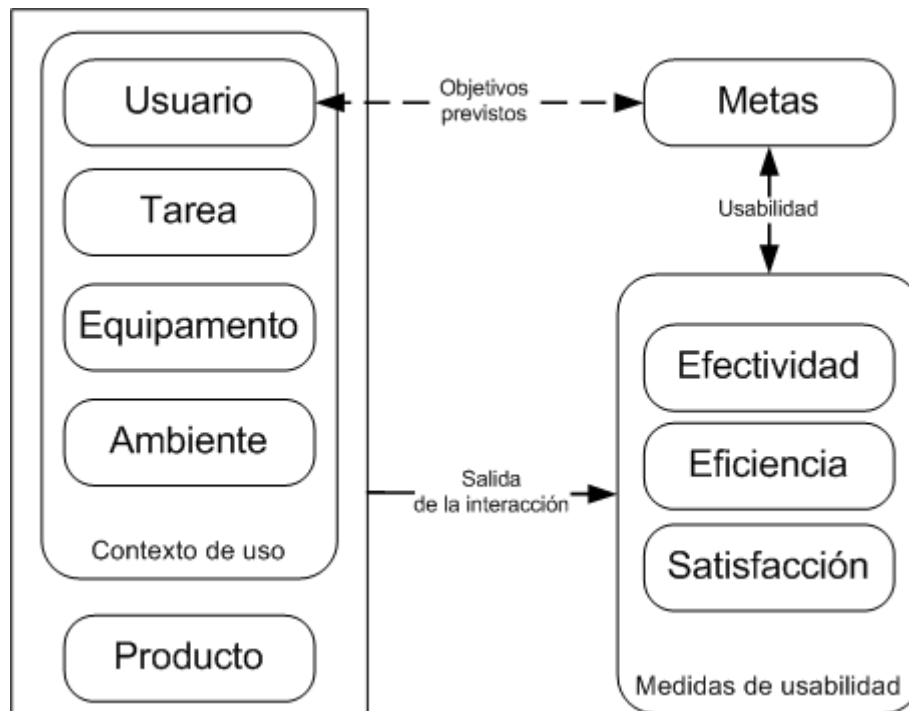


Ilustración 2 Framework de usabilidad de acuerdo a ISO 9241-11 [3]

El anterior cuadro es un resumen de la definición de usabilidad de la norma ISO 9241-11. A lo que se refiere es que el contexto de uso de un producto está formado por el propio usuario, la tarea que realiza, el equipamiento que utiliza para realizar la tarea y el ambiente en el que el usuario interactúa. Por ejemplo, un granjero (usuario) que desea plantar semillas (tarea) utilizando un arado (equipamiento) en una granja (ambiente). El producto en este caso sería la tierra.

Desarrollar el prototipo (Producir soluciones de diseño)

En esta actividad se desarrollará el prototipo de los escenarios seleccionados. Para esto, especificaremos el diseño en tres niveles:

- Diseño de la interacción
- Diseño de la información
- Diseño de la navegación

La salida de esta actividad es la *Solución de Diseño*.

Probar el prototipo con los usuarios (Evaluar diseño contra requerimientos)

Se revisa de manera sistemática la solución de diseño propuesta con el usuario con una modalidad de revisión adaptada para que el usuario guíe la misma. La salida de esta etapa es un *Reporte de Anomalías*.

Descripción detallada de cada etapa

Identificar las metas de usabilidad

En una primera instancia, se relevarán con los usuarios los requerimientos de usabilidad que guiarán las decisiones de diseño.

La usabilidad, de acuerdo a Quesenbery [20], tiene cinco dimensiones, a las que llama “Las cinco E”:

- Efectividad (Effective) – La exhaustividad y precisión con la que los usuarios logran sus objetivos.
- Eficiencia (Efficient) – La velocidad (con precisión) con la que este trabajo puede hacerse.
- Atractivo (Engaging) – Qué tan placentero, satisfactorio o interesante resulta el uso de una interfaz.
- Tolerante al error (Error tolerant) – Qué tan bien el producto previene los errores, y ayuda al usuario a recuperarse en caso de uno.
- Facilidad de aprendizaje (Easy to learn) – Qué tan bien el producto soporta tanto la orientación inicial como el aprendizaje en profundidad.

Es deseable atender a estos cinco atributos de manera uniforme. Sin embargo, hay que definir una prioridad entre estas dimensiones porque a menudo están interrelacionadas. Ser más estricto en un sentido puede resultar en degradar la calidad de otro atributo.

Por ejemplo, en una aplicación con formularios extensos, un error sin controlar puede llevar a pérdida de información y rechazo por parte del usuario si tiene que cargar todos los datos desde cero. Por otro lado, una funcionalidad o tarea que se realiza frecuentemente debería ser rápida de realizar.

La prioridad que le daremos a cada una de estas dimensiones vendrá dada por los objetivos propuestos.

Los autores Lauesen y Younessi [21] realizaron un estudio que recopila los estilos comunes de redacción de requerimientos de usabilidad en lenguaje natural. Los seis estilos son:

Rendimiento

Definiendo tareas y grupos de usuario en los que la usabilidad es importante, y métricas sobre el rendimiento esperado.

Ejemplo: Determinado porcentaje de un grupo de usuarios deberá finalizar la tarea correctamente en una cantidad de tiempo estipulada.

Defecto

Identificando defectos de usabilidad y poniendo una cota superior en ellos

Ejemplo: Determinado grupo de usuarios no deberá tener más que una cantidad acordada de inconvenientes para terminar una tarea concreta.

Proceso

Especificando requerimientos de usabilidad sobre los procesos y no sobre el negocio.

Ejemplo: Deberá haber por lo menos tres prototipos sobre determinada tarea y deberán ser validados con el usuario.

Este tipo de requerimiento no aplica, dado que está implícito en el proceso.

Subjetivo

Pone una meta en la satisfacción del usuario.

Ejemplo: Un determinado porcentaje de usuarios después de usar el sistema podrían calificarlo como satisfactorio.

Diseño

Especificando prototipos para definir los requerimientos de usabilidad

Ejemplo: El sistema deberá utilizar determinado control cuyo comportamiento será como el de otro sistema establecido .

Este tipo de requerimiento no aplica, dado que está implícito en el proceso.

Directriz

Definiendo un documento de lineamientos de usabilidad.

Ejemplo: Para los ingresos con una cantidad de valores posibles limitados, el usuario deberá poder seleccionar un valor de una lista.

Este tipo de requerimiento no aplica, dado que está implícito en el proceso.

Entonces tendremos los requerimientos especificados en una plantilla de esta manera:

- Identificación
- Dimensión de usabilidad que mide este requerimiento
 - Efectividad
 - Eficiencia
 - Atractivo
 - Tolerancia al error
 - Facilidad de aprendizaje
- Prioridad
 - Esencial
 - Condicional
 - Opcional
- Tipo de Redacción
 - Rendimiento
 - Defecto
 - Subjetivo
- Declaración del requerimiento
- Usuario o rol que lo validará

La salida de esta actividad es un documento con la lista de requerimientos de usabilidad como fue descrita anteriormente. A este artefacto lo llamaremos *Documento de Especificación de Requerimientos de Usabilidad*.

Identificar usuarios y tareas

En esta etapa necesitamos identificar a los usuarios del sistema y las tareas que necesitan completar.

Los usuarios son una parte de los interesados (stakeholders). Un interesado es un individuo que es materialmente afectado por el resultado de un proyecto.

En el estudio de la creación de un sitio web se utiliza mucho la técnica de relevamiento de *personas*. *Persona* es el término griego de máscara, y no se refiere al término en castellano de persona que es el que manejamos habitualmente.

La diferencia es que, en el diseño web se valen de esta técnica porque no conocen a la instancia de usuario real que va a utilizar su producto. En cambio, en la mayoría de los casos en una aplicación conocemos los usuarios concretos que van a usarla.

La *persona* sirve para dos motivos: uno, tipificar el objetivo de la audiencia que va a ser parte del proyecto, y otra guiar el diseño hacia esa audiencia. Si pudiésemos hacer un paralelismo entre las *personas* y nuestros objetivos, sería fácil de reconocer los roles que cumple un usuario en la organización. Ésta es una clasificación típica a la hora de especificar los usuarios.

Una serie de preguntas que nos podemos hacer para reconocer a los usuarios del sistema:

- ¿Quiénes van a ser los usuarios finales del sistema?
- ¿Quién va a ser afectado por la salida que produce el sistema?
- ¿Quién va a evaluar y aprobar los diseños que se producen con este proceso?
- ¿Hay usuarios externos cuyas necesidades deben ser atendidas?

Para tipificarlos, tendremos en cuenta las siguientes características, tomadas de las tres dimensiones de los usuarios de Nielsen [14]. La clasificación deberá ser de un conjunto de usuarios con un conjunto de habilidades similares y otras características que comparten los mismos roles y responsabilidades dentro del ámbito del sistema.

Usuario

- Nombre
- Rol
- Descripción general
- Conocimiento del tipo de sistema (Alto/Medio/Bajo)

Como conocimiento del sistema nos referimos a los mecanismos que se utilizan para convertir las entradas en salidas. Un tipo de sistema puede ser, por ejemplo, una aplicación web.

- Conocimiento del dominio (Alto/Medio/Bajo)

Como dominio nos referimos al dominio de la naturaleza de las tareas que este usuario típicamente desempeña, como puede ser un empleado que recién ingresó a una organización, o un gerente con muchos años de experiencia en una tarea.

- Conocimiento del medio (Alto/Medio/Bajo)

Como medio nos referimos al uso de, por ejemplo, una computadora, un navegador, etc. Un usuario puede conocer muy bien una tarea pero hacerla por otro medio, como por ejemplo, procesar y completar formularios en papel, pero no saber completar un formulario web.

Con respecto a las tareas tenemos las siguientes características:

Tareas

- Nombre de la tarea
- Objetivo de la tarea
- Requerimiento funcional relacionado
- Desglose de la tarea
- Frecuencia de uso de la tarea
 - Alta
 - Media
 - Baja

Se sugiere que para completar las categorías sugeridas, se tenga en cuenta la apreciación subjetiva del usuario, y se evidencie el porqué de la decisión. Este es un atributo importante dado que es un factor para poder priorizar en etapas posteriores.

A menudo	Alta
Con frecuencia	Alta
Frecuentemente	Alta
Muchas veces	Alta
Mucho	Alta
Siempre	Alta
Tantas veces	Alta
Todo el día	Alta
Todo el tiempo	Alta

A veces	Media
De vez en cuando	Media
Generalmente	Media
Varias veces	Media
Nunca	Baja

Los términos como cada año, cada día, todas las semanas o todos los días dependen fuertemente del uso general que se le da al sistema y queda a criterio de quien releva la tarea.

- Duración de la tarea
 - Alta
 - Media
 - Baja

Como en la frecuencia, la duración de la tarea es un atributo importante y se debe justificar la decisión de la clasificación objetivamente.

- Dependencias de la tarea

Las dependencias deberían ser otras tareas. Por ejemplo, si como parte de esta tarea es obligatorio asociar un elemento que se crea en otra, vamos a tener como dependencia esta otra tarea.

Para desglosar el contenido de la tarea, se propone una forma reducida de análisis de tareas [22]. El análisis de tareas es el estudio de lo que el usuario quiere hacer en términos de acciones o procesos cognitivos para lograr una tarea, proveyendo conocimiento sobre la tarea que el usuario quiere realizar.

El proceso de análisis de tareas tiene los siguientes pasos:

1. Identificar la tarea a ser analizada de una lista de tareas previamente relevadas.
2. Partir esta tarea entre 4 a 8 sub-tareas. Cada una de estas sub-tareas deben estar especificadas en términos de objetivos y entre ellos deberían cubrir toda el área de interés.
3. Decidir si este nivel de descomposición es suficiente o es necesario recabar más información. Hay que tener en mente que en esta etapa es necesario llegar a tener suficiente detalle como para poder especificar la interfaz con el usuario, modelar las interacciones y la navegación.
4. Volver al paso 2, ésta vez subdividiendo las sub-tareas.

5. Una vez terminado el proceso, mostrar el resultado a alguien que conozca bien la tarea pero no haya estado involucrado en el análisis.

Finalmente, para relacionar las tareas con los usuarios, usaremos una matriz simple donde en una dimensión tenemos los usuarios y en la otra tenemos las tareas. Podemos especificar las tareas en las que el usuario es un actor principal o frecuente y aquellas en las que el usuario es un actor secundario, solo forma parte de alguno de los pasos de la tarea o la salida de la tarea no lo afecta en gran medida.

La salida de esta actividad sería un documento con la lista de tareas, la lista de usuarios y la matriz de usuarios – tareas como fueron descritas anteriormente. A este artefacto lo llamaremos *Documento de Especificación de Contexto*.

Desarrollar el prototipo

La solución de diseño propuesta va a ser compuesta por diferentes partes, según la separación del diseño de la interfaz de usuario de Garrett antes citado [17]:

- El diseño de la interacción va a estar basado en el *Diccionario de elementos*, que describe en detalle de qué manera el usuario va a transferir al sistema las entradas.
- El diseño de la información, donde se van a definir las vistas que contendrán los elementos definidos en el diseño de la interacción.
- El diseño de la navegación, que va a estar basado en el Mapa de Sitio, donde se pueden visualizar a alto nivel las vistas y la interacción entre ellas.

Las especificaciones en conjunto van a conformar el artefacto llamado *Solución de diseño*.

Diseño de la interacción

Descripción

El diseño de la interacción consiste en seleccionar los elementos correctos para la tarea que el usuario quiere lograr y ordenarlos en la pantalla de manera que se entienda fácilmente y pueda ponerse productiva rápidamente [17].

Usamos los controles definidos en el anexo de **¡Error! No se encuentra el origen de la referencia.** Se pueden definir más controles de acuerdo a las necesidades, siguiendo la plantilla especificada.

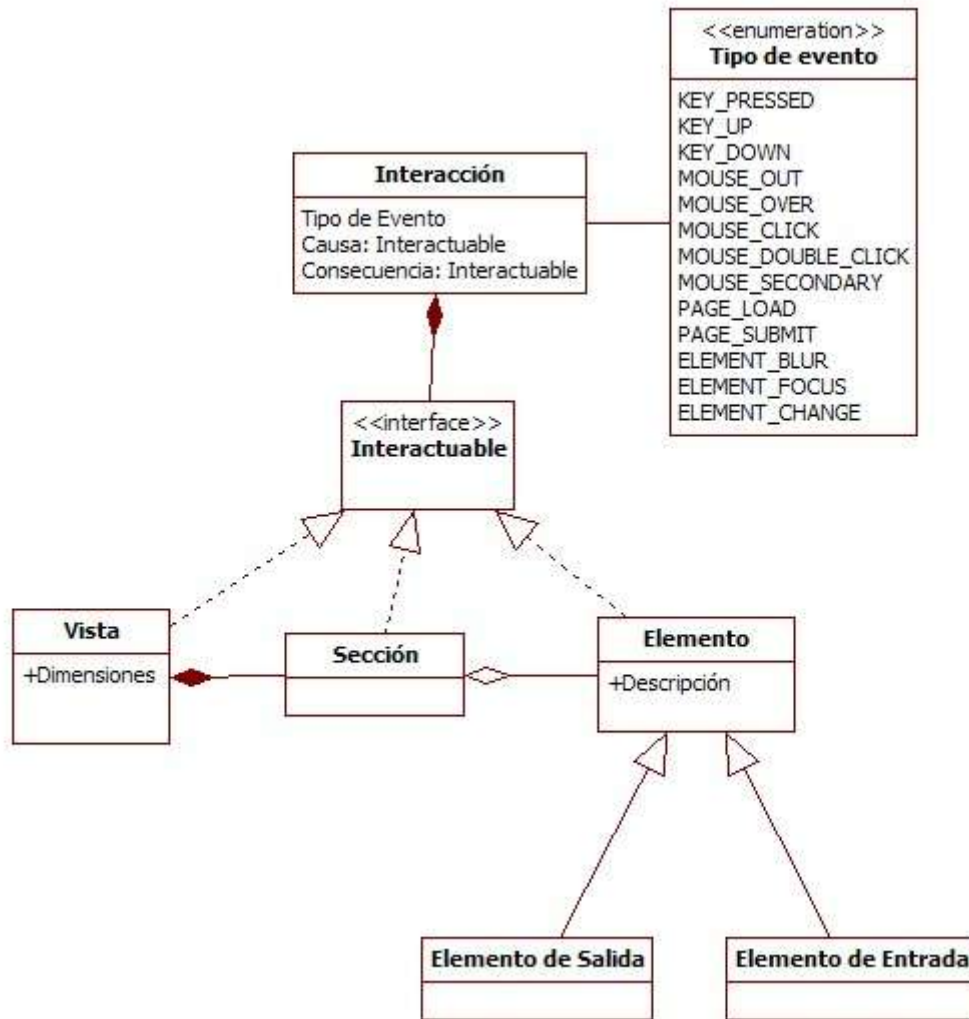
La definición de la interacción debería tener, como mínimo:

- Tipo de evento (Ver Anexo D: Definición de las interacciones)
- Causa
- Consecuencia

La causa y la consecuencia pueden ser una de estas dos cosas:

- Vista o Sección (Ver Anexo C: Notación de mapa de sitio)
- Elemento (Ver Anexo B: Diccionario de elementos)

El concepto de interacción se puede ver plasmado en el siguiente diagrama de clases:



Procedimiento

1. Recorremos la tarea, fijándonos si hay un indicio de interacción del usuario – sistema
2. Si encontramos interacción, buscamos el elemento adecuado para el estímulo (Ver Diccionario de elementos)
3. Agregamos el elemento a la lista de elementos con los siguientes datos:
 - a. ID - autoincremental
 - b. Código – Todo en mayúsculas, palabras separadas por guiones bajos, si el campo es optativo en cursiva
 - c. Tipo de elemento, seleccionado del Diccionario de elementos o sino, declarar un tipo de elemento para definirlo más adelante.
 - d. Descripción – Puede ser el texto literal que acompaña la etiqueta de un elemento o una descripción más extensa de por qué ese elemento fue seleccionado.

4. Para mayor claridad, estos elementos se pueden agrupar por sección
5. Para cada interacción, definirla siguiente manera:

Tipo de evento (Ver Anexo C: Definición de las interacciones)

- a. Causa (Elemento , Sección o Vista)
- b. Consecuencia (Elemento, Sección o Vista)
 - i. Condición en la cual se da la consecuencia
6. Si quedo algún elemento sin definir, definirlo ahora con la plantilla del Diccionario de elementos

La salida de esta parte de la actividad sería un documento con la lista de elementos como fueron descritos anteriormente y el diccionario de elementos nuevos definidos. A este artefacto lo llamaremos *Documento de Especificación de la Interacción*.

Diseño de la información

Descripción

Los elementos especificados anteriormente tienen que ser organizados visualmente en la pantalla. Por lo tanto, necesitamos ubicarlos en lo que llamaremos una vista.

La vista es una representación visual de la disposición de los elementos en una pantalla.

Los datos que debe tener esta vista son, como mínimo:

- Nombre de la vista
- Dimensiones (pueden ser expresadas en píxeles, mm, cm)
- Imagen con la disposición relativa de los elementos, etiquetados

En ocasiones es necesario dar cierta realimentación al usuario sobre la interacción. A veces esto es una ventana emergente o una sección de la pantalla que indica el error, advertencia o información adicional. Por lo tanto, se sugiere agregar además de una vista con la pantalla, como si el usuario la estuviera cargando por primera vez: una vista que ejemplifique cómo se van a mostrar estos mensajes.

Procedimiento

1. Crear la imagen de la vista con el tamaño suficiente para disponer de los elementos que contiene. No tiene que ser necesariamente en escala 1:1, pero las proporciones deben mantenerse.
2. Recorrer la Lista de elementos generada en el Diseño de la interacción. Agregar la representación gráfica del elemento a la imagen.
3. Agregar una llamada al elemento con el número de ID correspondiente a la Lista de elementos. Una llamada es un número dentro de un círculo resaltado. Si la vista está compuesta por más de una sección, resaltar las llamadas de las mismas secciones del mismo color.
4. Chequear que la tarea pueda ser recorrida con la pantalla de la manera actual. De lo contrario, volver a la definición de los elementos o arreglar nuevamente los controles en la vista, según corresponda.

La salida de esta parte de la actividad sería un documento con las vistas como fueron descritas anteriormente. A éste artefacto lo llamaremos *Documento de Especificación de la Información*.

Diseño de la navegación

Descripción

El diseño de la navegación tiene que cumplir tres objetivos [17]:

- Proveer a los usuarios con los medios para recorrer la aplicación
- Comunicar la relación con los elementos que contiene
- Comunicar la relación entre sus contenidos y la vista actual

Mapa de sitio

Un mapa de sitio es una manera de visualizar páginas representativas en una aplicación que se puede beneficiar de identificar páginas, vistas, estados e instancias de lo que se debe mostrar [23]. No confundir con el mapa de sitio como herramienta alternativa de navegación, donde los usuarios tienen vínculos a todas las páginas de un sitio y éstas están organizadas por contenido.

Debe cumplir con dos condiciones para resultar útil:

- Debe ser una **taxonomía**, porque queremos clasificar pantallas por su contenido
- Debe ser **jerárquica**, para organizar el contenido desde la mayor abstracción a la materialización concreta del contenido

La notación a utilizar en el mapa de sitio está especificada en Anexo C: Notación de mapa de sitio.

Conector

El conector es el elemento que permite la unión de navegación entre dos vistas. La acción de un conector debe ser detallada por separado, y debería tener como mínimo:

- Identificador del conector
- Vistas que vincula
- Identificador de la interacción que dispara la navegación

Procedimiento

1. Para todos los vínculos de la Lista de elementos, definir un conector.
2. Para todos los botones que tengan acción de navegación en la Lista de elementos, definir un conector.
3. Una vez definidos todos los conectores, dibujar todas las vistas relacionadas y sus conexiones. Ver Notación de mapa de sitio para detalle de la notación.

Probar el prototipo con los usuarios

Para probar la solución de diseño, se adapta un proceso de recorrida del estándar IEEE Std. 1028-1997 [24].

Introducción

Una recorrida (walkthrough) es una actividad de revisión por pares en el que un programador o diseñador conduce miembros del equipo de desarrollo y otras partes interesadas a través de un producto de software, mientras los participantes hacen preguntas y comentarios acerca de los posibles errores, la violación de las normas de desarrollo, y otros problemas. El propósito de esta actividad es evaluar y validar el diseño producido en la anterior etapa.

Una variación de la revisión es la revisión inversa. En ese caso, el usuario recorre el producto de software y describe los pasos para llevar a cabo una tarea. La ventaja que esto presupone es que el usuario valida la solución de diseño, justificando que decisión toma en cada paso.

El propósito de esta actividad es evaluar el diseño producido en la etapa anterior. Sus objetivos son:

- Encontrar anomalías
- Mejorar el diseño
- Considerar implementaciones alternativas
- Evaluar adherencia a especificaciones y estándares

Responsabilidades

Los siguientes roles son establecidos para la recorrida:

- a) Líder
- b) Registrador
- c) Autor
- d) Usuario

a) Líder

El líder debe conducir el recorrido, manejar las tareas administrativas pertinentes al recorrido (distribuir documentaciones y coordinar la reunión) y asegurarse de que la reunión se lleva de manera ordenada. El líder deberá preparar una declaración de los objetivos para guiar al equipo en el recorrido. También debe limitar a un ítem de acción o

decisión determinado para cada punto de discusión. Finalmente, deberá expedir la salida de esta actividad.

b) Registrador

El registrador deberá anotar todas las decisiones y acciones identificadas que ocurran en la reunión. Adicionalmente, debe registrar todos los comentarios respecto de las anomalías encontradas, cuestiones de estilo, omisiones, contradicciones, oportunidades de mejora o enfoques alternativos.

c) Usuario

El usuario debe presentar el diseño en el recorrido.

Entrada

Las entradas del recorrido son:

- a) Declaraciones de los objetivos para el recorrido
- b) La tarea a ser recorrida con el diseño
- c) El diseño a ser examinado
- d) Los estándares en vigencia en relación al diseño
- e) Categorías de anomalías (Ver Anexo A - Clasificación de anomalías)

Procedimiento

1. El autor hace un resumen del objetivo de la solución de diseño a presentar.
2. Cada miembro del equipo debe estar preparado para la reunión, habiendo repasado el material de entrada provisto por el líder previamente.
3. El líder presenta a los participantes y describe sus roles. Debe recordar que hay que enfocarse en la detección de anomalías y no en su resolución. También debe recordar que deben comentar sobre el prototipo y no sobre su autor. Los miembros del equipo pueden preguntarle al autor sobre el prototipo. El autor no debe inducir de ninguna manera la manera de llevar a cabo el flujo de la pantalla.
4. Discusión general, basada en los ítems que generaron antes la reunión.
5. Discusión detallada, donde el usuario recorre en detalle el prototipo y los mapea con los pasos de la tarea a seguir.

Durante la reunión:

1. El autor o el líder deben hacer un resumen en modo de presentación sobre el prototipo bajo revisión.

2. El líder debe coordinar la discusión sobre las inquietudes generales.
3. El usuario, guiándose con la tarea, debe describir, en voz alta, cómo mapea un paso lógico de ésta con una acción con el prototipo.
4. Los miembros del equipo deben notar anomalías específicas cuando el usuario llega a una parte del prototipo relacionada.
5. El registrador debe anotar las recomendaciones y acciones que devienen de la discusión de las anomalías.

Después de la reunión, el líder debe emitir la salida de la reunión de acuerdo a lo explicado en la sección Salida.

Salida

La salida de la reunión debe tener, como mínimo, los siguientes puntos:

- Los miembros del equipo y sus roles
- El producto examinado
- Una declaración de los objetivos que debían ser cumplidos durante la reunión y si fueron cumplidos.
- Una lista de recomendaciones hechas sobre cada anomalía
- Una lista de acciones, fechas límite y responsables
- Las recomendaciones hechas por el equipo de trabajo sobre cómo disponer de las deficiencias y anomalías no resueltas
- Las propuestas hechas por el equipo de trabajo sobre el seguimiento de las anomalías

La salida de esta actividad conformará el documento llamado *Reporte de anomalías*.

Capítulo 5 Caso de estudio

Descripción

Se va a estudiar un caso típico de interacción con el usuario: Un foro de debate. Se espera mediante este escenario recorrer todas las actividades del proceso y observar las salidas para verificar cómo se plasman los objetivos de los usuarios en la especificación.

Un foro de debate o tablero de mensajes es un sitio de debate en línea en el cual la gente puede tener conversaciones en forma de mensajes publicados. Este foro de debate es sobre derecho. Si bien hay muchos usuarios que no están familiarizados con este tipo de aplicaciones, son los que pueden enriquecer más los debates con su conocimiento (por ejemplo, abogados con mucha trayectoria que nacieron mucho antes del boom de las computadoras personales y todavía usan la máquina de escribir para redactar escritos). Por otro lado, hay estudiantes de derecho que van al sitio buscando información y opinión sobre jurisprudencia. Estos usuarios sí conocen las herramientas y en su mayoría han utilizado un foro de debate anteriormente.

Requerimientos funcionales (parciales)

- El usuario podrá registrarse en el sitio.
- El sitio tendrá foros de debate con distintos temas
- El usuario contará con la posibilidad de crear un nuevo tema de debate en un foro seleccionado.

Documento de especificación de requerimientos de usabilidad

Objetivos de usuario

- Los usuarios quieren hacer bien la tarea en el entorno de aprendizaje.
- Los usuarios quieren expresar su opinión de un tema rápidamente.

Tipo de usuarios

- Tipo I: Tienen experiencia en herramientas informáticas y navegación web. Han usado foros de debate en línea de otros proveedores anteriormente.
- Tipo II: Tienen experiencia en herramientas informáticas y navegación web. No han usado foros de debate en línea anteriormente.

Metas de usabilidad

Meta 1: En un grupo de usuarios tipo I, el 80% deberá terminar la tarea correctamente.

Meta 2: En un grupo de usuarios tipo II, el 60% deberá terminar la tarea correctamente.

Meta 3: El 80% de los usuarios deberá crear un nuevo tema de debate, que contenga no más de 50 palabras, en menos de cinco minutos.

Meta 4: El 80% de los usuarios deberá declararse conforme o muy conforme al preguntarle por la facilidad de uso de la herramienta para crear nuevos mensajes

Meta 5: No más del 20% de los usuarios de Tipo II deberá declarar en una encuesta que tuvo dificultades para utilizar la herramienta para crear nuevos mensajes por primera vez.

Meta 6: El 90% de los usuarios tienen que completar la tarea solamente con la información que proporciona el sistema

Meta 7: No más del 20% de los encuestados deber responder afirmativamente a la pregunta sobre si el usuario al ser ofrecido entre ver el mensaje creado o ir a ver otros mensajes, elige ver el mensaje creado para saber si se creó bien.

Matriz

ID	Nombre	Dimensión					Prioridad			Tipo			Usuario
		Efectividad	Eficiencia	Acoplamiento	Tolerancia al error	Facilidad aprendizaje	Esencial	Condicional	Opcional	Rendimiento	Defecto	Subjetivo	
1	Meta 1	X					X			X			Tipo I
2	Meta 2	X					X			X			Tipo II
3	Meta 3		X					X		X			Todos
4	Meta 4			X			X				X		Todos
5	Meta 5					X	X				X		Tipo II
6	Meta 6				X		X			X			Todos
7	Meta 7			X				X			X		Todos
Total		33%	17%	17%	13%	9%	83%	17%	0%				

Documento de especificación de contexto

Tarea 1: Registrarse en el sitio

Flujo normal

- El usuario ingresa
 - o Nombre
 - o Apellido
 - o Correo electrónico
 - o Confirmación del correo electrónico
 - o Contraseña
 - o Sexo (Hombre/Mujer)
 - o Día, mes y año de su fecha de nacimiento
- El usuario confirma enviar la información que completó
- El usuario ingresa las **palabras que observa** en una **imagen dinámica**
- El usuario confirma **enviar** la información que completó

Validaciones

- CAMPOS_OBLIGATORIOS: Todos los campos son obligatorios
- CORREO_INVALIDO: El **correo electrónico** debe ser válido
- CORREO_NO_COINCIDENTE: **Correo electrónico** y **confirmación del correo electrónico** coinciden
- CONTRASEÑA_CORTA: La **contraseña** debe tener una longitud mínima de 6 caracteres
- CAPTCHA_COINCIDE: El texto de la imagen dinámica y el texto que completó el usuario deben coincidir

Tarea 2: Crear un nuevo tema de debate

Precondición

- El usuario debe estar registrado y autenticado en el sitio

Flujo normal

- El usuario selecciona un foro para visualizar sus temas. Los foros están divididos en categorías y listados de la siguiente manera:
 - o ¿Tiene un mensaje nuevo desde la última vez que el usuario visitó el foro? (Si/No)
 - o Título del foro
 - o Descripción del foro

- Cantidad de temas
- Cantidad de mensajes
- Último mensaje
 - Título del tema
 - Fecha
 - Usuario que lo hizo
- El sistema muestra
 - Título del foro
 - La posibilidad de crear un nuevo mensaje
 - La navegación dentro del foro
 - Cuadro de búsqueda rápida
 - Una lista paginada de temas con los siguientes datos:
 - ¿Tiene un mensaje nuevo desde la última vez que el usuario visitó el foro? (Si/No)
 - Título del tema
 - Descripción del tema
 - Navegación de las páginas del tema, si tiene más de una
 - Cantidad de respuestas en el tema
 - Autor
 - Lecturas
 - Último mensaje
 - Título del tema
 - Fecha
 - Usuario que lo hizo
 - Lista desplegable con los otros foros para navegación rápida
- El sistema despliega los siguientes campos para ser completados:
 - Asunto
 - Descripción
 - Cuerpo del mensaje
- El usuario completa los campos y confirma los datos ingresados
- El sistema crea el mensaje en el foro seleccionado

Validaciones

- El mensaje tiene que tener más de N caracteres, siendo N un parámetro a definir por el administrador (LONGITUD_MENSAJE_INSUFICIENTE)
- Todos los campos del mensaje son obligatorios (CAMPOS_REQUERIDOS)

Solución de diseño

Diseño de la interacción

Lista de foros

ID	Nombre	Tipo	Descripción
1	MENSAJE_NUEVO	Indicador ON/OFF	
2	TITULO_FORO	Vínculo	Título del foro
3	DESCRIPCION_FORO	Etiqueta/Texto	Descripción del foro
4	CANTIDAD_TEMAS	Etiqueta/Número	Cantidad de temas del foro
5	CANTIDAD_MENSAJES	Etiqueta/Número	Cantidad de mensajes de los temas del foro
6	ULTIMO_MENSAJE		
6.1	TITULO_TEMA	Vínculo	Título del tema
6.2	FECHA_ULTIMO_MENSAJE	Etiqueta/Fecha	Formato - Hoy/Ayer/Lu ...Dom - "a las" - hh:mm (am/pm)
6.3	USUARIO	Vínculo	Nombre del usuario del último mensaje

Temas

ID	Nombre	Tipo	Descripción
1	TITULO_FORO	Título	Título del foro
2	NUEVO_TEMA	Botón	Crear nuevo tema
3	NAV_TEMAS	Miga de pan	Sitio > Foro
4	CUADRO_BUSQUEDA		
4.1	BUSQUEDA_RAPIDA	Texto	Cuadro de texto para la búsqueda rápida
4.2	BUSCAR	Botón	Botón para buscar
5	NAV_FOROS	Menú	Otros foros para navegar

Interacción

Tipo de evento: MOUSE_CLICK

Causa:

- Elemento: NUEVO_TEMA

Consecuencia:

- Vista: VISTA_NUEVO_TEMA

Tipo de evento: MOUSE_CLICK

Causa:

- Elemento: BUSCAR

Consecuencia:

- Vista: VISTA_BUSQUEDA (No relevada)

Lista de temas

ID	Nombre	Tipo	Descripción
1	MENSAJE_NUEVO	Indicador ON/OFF	
2	TITULO_TEMA	Vínculo	Título del tema
3	DESCRIPCION_TEMA	Etiqueta	Descripción del tema
4	NAV_PAG_MENSAJES	Navegación de tema	[Ir a la página: 1 ... N]
5	RESPUESTAS	Etiqueta/Número	Cantidad de respuestas del tema
6	AUTOR	Vínculo	Nombre del autor
7	LECTURAS	Etiqueta/Número	Cantidad de lecturas del tema
8	ULTIMO_MENSAJE		
8.1	FECHA_ULTIMO_MENSAJE	Etiqueta/Fecha	Formato <ul style="list-style-type: none"> - Hoy/Ayer/Lu ...Dom - "a las" - hh:mm (am/pm)
8.2	USUARIO	Vínculo	Nombre del usuario del último mensaje

Diseño de la información

1: INICIO

Foros

	Temas	Mensajes	Último mensaje
<input type="radio"/> Derecho Civil - Comercial Descripción de éste foro	254	12314	Holaaaa Hoy a las 11:43 AM por Fidim
<input checked="" type="radio"/> Derecho Penal Descripción de éste foro	1234	54353	Holaaaa Hoy a las 11:43 AM por Fidim
<input type="radio"/> Jovenes abogados Descripción de éste foro	3465	23245	Holaaaa Hoy a las 11:43 AM por Fidim
<input type="radio"/> Mediación Descripción de éste foro	15	1123	Holaaaa Hoy a las 11:43 AM por Fidim
<input type="radio"/> Derecho previsional Descripción de éste foro	2342	12345	Holaaaa Hoy a las 11:43 AM por Fidim

2: FORO

Derecho Civil - Comercial

[Inicio](#) > [Derecho Civil - Comercial](#)

	Respuestas	Autor	Lecturas	Último mensaje
<input type="radio"/> Consulta anticresis Ir a la página 1 2	254	Cordo	12314	Hoy a las 11:43 AM por Fidim
<input type="radio"/> Sucesión auto + inmueble Descripción de éste tema	1234	Coco	54353	Hoy a las 11:43 AM por Fidim
<input type="radio"/> Acuse de negligencia	3465	Javi	23245	Hoy a las 11:43 AM por Fidim
<input type="radio"/> Plazo prescripción	15	Fede	1123	Hoy a las 11:43 AM por Fidim
<input type="radio"/> Caducidad en ejecutivo	2342	Pato	12345	Hoy a las 11:43 AM por Fidim

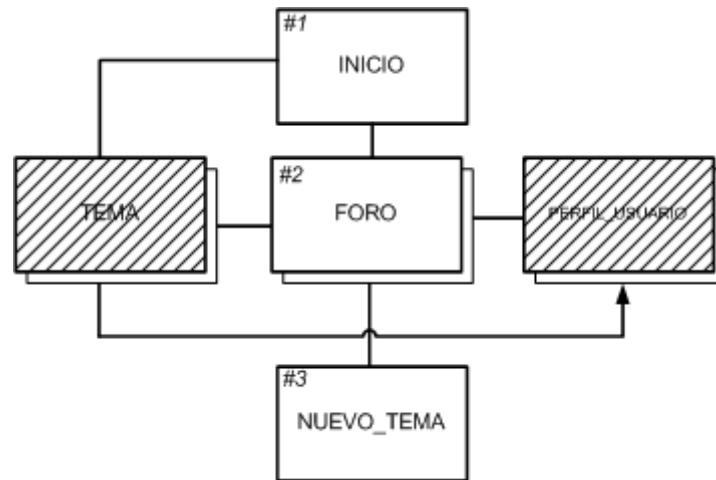
3: NUEVO_TEMA

The image shows a web form titled "Nuevo tema" with five numbered annotations in yellow circles:

- 1: Points to the title "Nuevo tema".
- 2: Points to the "Asunto" input field containing the text "Presentación".
- 3: Points to the "Descripción" input field.
- 4: Points to the text "Hola...soy nuevo en el foro..." inside the description field.
- 5: Points to the "Enviar" button.

Diseño de la navegación

Mapa de sitio



Conectores

ID	Vista	Vinculo	Destino
1	INICIO	TITULO_FORO	FORO
2	INICIO	TITULO_TEMA	TEMA
3	INICIO	USUARIO	PERFIL_USUARIO
4	FORO	TITULO_TEMA	TEMA
5	FORO	AUTOR	PERFIL_USUARIO
6	FORO	USUARIO	PERFIL_USUARIO

Diseño de la interacción

Elementos

ID	Nombre	Tipo	Etiqueta
1	NOMBRE	Texto	Nombre
2	APELLIDO	Texto	Apellidos
3	EMAIL	Texto	Tu correo electrónico
4	EMAIL_CONFIRMAR	Texto	Vuelve a escribir tu correo
5	CONTRASEÑA	Texto oculto	Contraseña
6	SEXO	Menú sexo	Sexo
7	FECHA	Menú fecha	Fecha de nacimiento
8	REGISTRAR	Botón	Regístrate

Interacción

Tipo de evento: MOUSE_CLICK

Causa:

- Elemento: REGISTRAR

Consecuencia:

- Vista: REGISTRO_CAPTCHA
- Condición: Todos los campos del área REGISTRAR validan correctamente
- Vista: REGISTRO_ERROR
- Condición: alguna de las condiciones de validación no se cumplieron

Elementos

ID	Nombre	Tipo	Etiqueta
1	CAPTCHA	Captcha	
2	TEXTO_CAPTCHA	Texto	Texto que se muestra en la imagen
3	REGISTRAR	Botón	Regístrate
4	VOLVER	Vínculo	< Volver

Eventos

Tipo de evento: MOUSE_CLICK

Causa:

- Elemento: REGISTRAR

Consecuencia:

- Vista: REGISTRO_EXITOSO
- Condición: Todos los campos del área REGISTRAR_CAPTCHA validan correctamente

- Vista: REGISTRO_CAPTCHA_ERROR
- Condición: alguna de las condiciones de validación no se cumplieron

Tipo de evento: MOUSE_CLICK

Causa:

- Elemento: VOLVER

Consecuencia:

- Vista: REGISTRO_DATOS

Elementos definidos

Menú sexo

Valores:

- **Selecciona el sexo:**
- Mujer
- Hombre

Menú fecha

El menú fecha está compuesto por tres menús

Menú día

Valores:

- **Día:**
- Valores desde el 1 al 31

Menú mes

Valores

- **Mes:**
- Valores desde enero hasta diciembre

Menú año

- **Año:**

- Valores desde 2011 hasta 1905

Validación:

- Cuando se selecciona un mes donde el día seleccionado anteriormente es inválido (por ejemplo, se selecciona febrero y previamente se había seleccionado el día 31) el menú de día vuelve al valor por defecto.

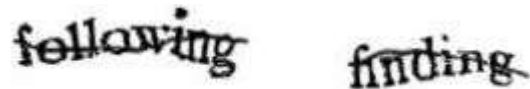
Captcha

El captcha es una prueba para asegurarse que la respuesta no es generada por una computadora.

Imagen

Valor: Aleatoriamente crea dos palabras distorsionadas

Ejemplo



following finding

Diseño de la información

1: REGISTRO_DATOS

A registration form with the following elements and numbered callouts:

- 1: Text input field for "Nombre:"
- 2: Text input field for "Apellidos:"
- 3: Text input field for "Tu correo electrónico:"
- 4: Text input field for "Vuelve a escribir tu correo:"
- 5: Text input field for "Contraseña:"
- 6: Dropdown menu for "Sexo:" with the text "Selecciona el sexo:"
- 7: Three dropdown menus for "Fecha de nacimiento:" labeled "Dia:", "Mes:", and "Año:"
- 8: "Registrate" button

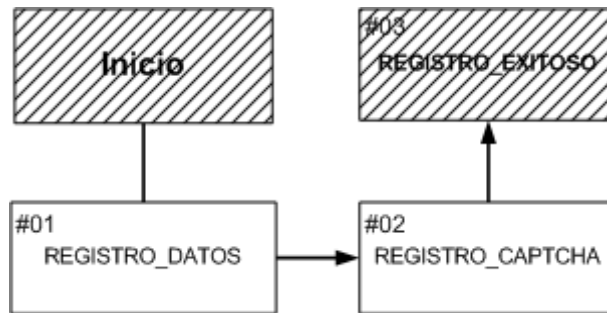
2: REGISTRO_CAPTCHA

A CAPTCHA registration form with the following elements and numbered callouts:

- 1: A box containing the distorted text "fi dim babo"
- 2: Text input field for "Texto que se muestra en la imagen:"
- 3: "< Volver" link
- 4: "Registrate" button

Diseño de la navegación

Mapa de sitio



Conectores

Interacción: CLICK_REGISTRAR

Vista desde: REGISTRO_DATOS

Vista hasta: REGISTRO_CAPTCHA

Reporte de anomalías

El líder completa una planilla con las anomalías detectadas.

ID	Descripción	Síntoma	Tipo	Usabilidad	Responsable
1	No debería ser obligatorio preguntar el sexo para registrarse	Entrada - Parámetros incompletos o faltantes	Problema de documentación - Ítem incompleto	Degradada	Analista
2	No hay criterio de búsqueda en la búsqueda rápida	Entrada - Descripciones incorrectas o faltantes	Problema de documentación - Ítem confuso	Afectada	Analista
3	Se deberían separar los campos de email en el registro para que sea más intuitivo	Entrada - Entrada incorrecta aceptada	Mejora	Afectada	Diseñador
4	La descripción debería estar marcada como optativa	Entrada - Entrada correcta no aceptada	Requerimientos - Mejorar usabilidad	No afectada	Diseñador

Conclusiones del caso de estudio

Con este caso de estudio pudimos observar la manera de la cual el usuario y el equipo interactúan en el proceso, generando los artefactos. Se ilustró únicamente una iteración, pero este proceso es netamente iterativo; dado el impacto de las anomalías pueden pasarse a otro ciclo o entregar los requerimientos con observaciones.

Conclusión

Aunque el diseño y el desarrollo idealmente iteran, la realidad dicta que generalmente estas dos etapas se separan. Por lo tanto, muchas veces los diseños se cambian sin aporte del diseñador, sin un conocimiento profundo del aporte que el diseño hace a los objetivos del usuario y sin el fundamento de las decisiones de diseño.

Los diseños siempre cambian a lo largo del proceso de desarrollo por diversas razones que no es el caso analizar aquí. La documentación de diseño ayuda a que el producto final se corresponda con los objetivos de usuario originales e informe sobre los cambios requeridos. También es capaz de capturar todas las decisiones y fundamentos tomados para que el resto de los miembros del equipo los tenga presentes.

Es responsabilidad de la organización que implemente estos artefactos decidir el nivel de detalle en el cual se apega a estas especificaciones de acuerdo a la realidad que dictan sus tipos de clientes y proyectos.

Como comentario final, podemos concluir que la aplicación de estos artefactos tiene los siguientes beneficios:

- El proceso, al estar basado en HCD, asegura que los requerimientos funcionales y de usabilidad detrás de la interfaz de usuario y el diseño de interacción son plasmados en el producto final.
- El diseño es iterativo y tiene como prioridad las metas de los usuarios.
- La documentación puede ser utilizada tanto como entradas para la elaboración del software como para comunicar a los interesados sobre el refinamiento de los requerimientos.

Anexo A Diccionario de elementos

La descripción de un elemento incluye, como mínimo:

- Identificador del elemento
- Descripción

Interacción por defecto (Ver Definición de las interacciones)

- Imágenes de los estados
- Instancias comunes

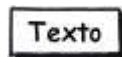
Controles de entrada

Botón

Descripción: Un rectángulo con una leyenda y/o una imagen.

Instancias comunes:

- Aceptar: Confirmar una acción y navegar a otra pantalla
- Cancelar: Abortar una acción sin guardar cambios y navegar a otra pantalla
- Aplicar: Confirmar acciones, pero no navegar a otra pantalla



Casilla de verificación

Descripción física: Es un cuadrado vacío.

Descripción de la interacción

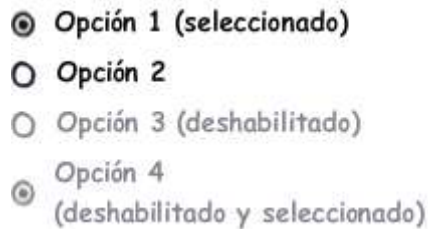
- Estados: Seleccionado o no seleccionado.
- Métodos de ingreso: Puntero y teclado

Una casilla de verificación es un elemento de interfaz que permite al usuario seleccionar múltiples opciones de una lista.

- No seleccionado
- Seleccionado
- Deshabilitado
- Deshabilitado Seleccionado

Botón de radio

Un “botón de radio” es un control que permite que solo una de las opciones posibles esté seleccionada al mismo tiempo. Su nombre viene de las radios de auto viejas que, al presionar uno de los botones, el que estaba seleccionado anteriormente salía para afuera.



Deslizador

Un deslizador es un elemento que permite al usuario deslizar, de arriba abajo o izquierda derecha para seleccionar un posible valor. Este control posibilita visualmente elegir entre un rango de valores posibles, y pueden estar marcados algunos valores significativos sobre el deslizador o estar acompañado de un texto (o una entrada de texto) para ir marcando el valor seleccionado o ingresarlo a mano.



Desplazador

Un desplazador es un elemento que permite al usuario seleccionar un valor numérico presionando botones para incrementar o disminuirlo.



Menus

Un menú es un control que permite al usuario seleccionar un elemento

Encarnaciones comunes:

- Select (Se muestra una sola opción, al presionar en el control se despliegan las opciones posibles)
- Select múltiple (Se muestran muchas opciones, se pueden elegir varias)
- Combo box (Se puede tipear un valor o elegir uno de una lista)



Entrada de texto

El campo de texto es la expresión más libre que puede tener un usuario respecto del ingreso. Es un control en el que se puede ingresar un texto sin límite de longitud.

Encarnaciones comunes:

- Texto común
- Texto oculto (por ejemplo, para contraseñas)
- Texto en área, para escribir detalles o descripciones



Selección de archivo

Permite seleccionar un archivo en la máquina del cliente, independientemente del sistema operativo.

Calendario

Permite seleccionar una fecha. Hay que tener especial cuidado con la fecha predefinida, como por ejemplo, cuando se tiene que ingresar una fecha de nacimiento y por defecto el control carga la fecha de hoy.



Vínculo

El vínculo lo que hace es, justamente, encadenar un documento con otro. Esto es parte del comportamiento inherente de la web como un protocolo de hipertexto. Generalmente es un texto de distinto color al resto, subrayado. El vínculo impacta en el diseño de la navegación, dado que define un conector, así como los botones que accionan una navegación.

Vínculo

Controles de salida

Listado

Es común que en las aplicaciones aparezca una manera tabular de mostrar una búsqueda.

Las características comunes de estos listados son:

- Hacer clic en el encabezado para ordenar por ese criterio
- Arrastrar una columna para cambiar el orden
- Arrastrar el borde de una columna para cambiar su ancho
- Colores alternados para columnas

Nombre ▲	Edad	Seudon	¿Asiste?
Patricio Fagalde	27	Fidim	<input checked="" type="checkbox"/>
Leandro Ferrigno	25	Rada	<input checked="" type="checkbox"/>
Lucas Lopez Elias	26	Luqi	<input checked="" type="checkbox"/>
Alejandro Molinari	31	Ale	<input checked="" type="checkbox"/>
Javier Monti	30	Monchi	<input type="checkbox"/>

Título

Un título es un encabezado al principio de una vista o sección. Sirve para indicar al usuario el lugar en el que está ubicado o qué contenido está visualizando en ese momento.

Título

Etiqueta

Una etiqueta es lo que acompaña a un control (de la otra sección) para clarificar el propósito de éste. Para clarificar, se especifica qué tipo de dato se muestra, como por ejemplo:

- Texto
- Número
- Fecha (dd/mm/aaaa)

Etiqueta

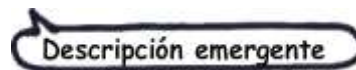
Barra de progreso

Para un proceso que toma mucho tiempo o varios pasos, la barra de progreso da un sentido de avance respecto de los pasos que se van haciendo.



Descripción emergente

Es un elemento que muestra una ayuda adicional al usuario al posar el puntero sobre un control.



Anexo B Definición de las interacciones

Tipo de evento

Código	Nombre	Descripción
Teclado		
KEY_PRESSED	Tecla o combinación de teclas	Cuando la tecla o combinación de teclas fue presionada y levantada
KEY_UP	Tecla levantada	Cuando la tecla dejó de estar presionada
KEY_DOWN	Tecla presionada	Cuando la tecla fue presionada
Puntero		
MOUSE_OVER	De adentro hacia afuera	Cuando el puntero salió del área de interacción del elemento
MOUSE_OUT	De afuera hacia adentro	Cuando el puntero entró en el área de interacción del elemento
MOUSE_CLICK	Clic	El usuario hizo clic en el elemento
MOUSE_DOUBLE_CLICK	Doble Clic	El usuario hizo doble clic en el elemento
MOUSE_SECONDARY_CLICK	Clic secundario	El usuario hizo clic con el botón secundario
Vista		
VIEW_LOAD	En la carga	Al cargarse la vista
VIEW_SUBMIT	En el envío	Al enviar datos a través de un

		formulario
Elemento		
ELEMENT_BLUR	Perder foco	El elemento perdió foco, o sea, estaba seleccionado para ingresar datos pero ahora no lo está más
ELEMENT_FOCUS	Ganar foco	El elemento ganó foco, o sea, fue seleccionado para ingresar datos
ELEMENT_CHANGE	Cambio	El elemento cambió su valor

Anexo C Notación de mapa de sitio

Notación general

Si un elemento aún no ha sido relevado o está fuera del alcance de la demostración, aparecerá con un fondo rayado en diagonal.

Vista

Una vista es la forma más básica de representación en el diseño de navegación. Está representada por un rectángulo, con un nombre representativo y una referencia de prototipo (de la sección anterior).

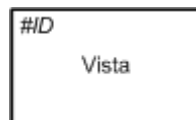


Ilustración 3 Notación de una Vista

Sección

Una sección es una parte dentro de una vista que es modelada por separado, sea por su complejidad, para simplificar el proceso o porque el ciclo de aprobación va por separado. El nombre de la sección debe ser único porque puede ser utilizado en distintas vistas, pero no se le agrega un ID para que no haya una sobrecarga de notación. Se utiliza para clarificar el contenido de una vista sin tener que ir al detalle del prototipo.

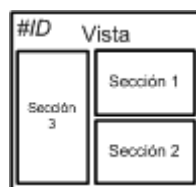


Ilustración 4 Notación de una Sección

Plantilla

Una plantilla es una vista cuyo contenido varía, pero su estructura no. Para citar un ejemplo, podría ser una vista de edición de varias entidades (la información, el contenido de la entidad, cambia, pero la estructura de la página no). Se representa como una pila de vistas.

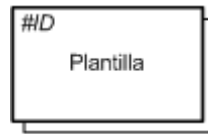


Ilustración 5 Notación de una Plantilla

Conector

Un conector indica la navegación entre dos componentes en el mapa de sitio. La punta de la flecha indica si la navegación es en un solo sentido; si no tiene flechas, la navegación es bidireccional.

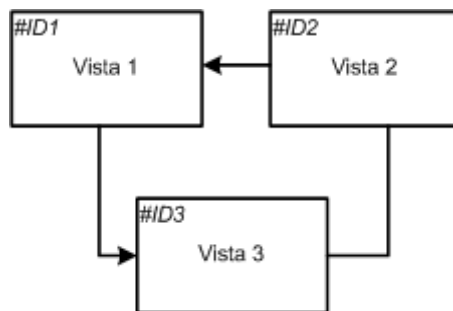


Ilustración 6 Notación de un Conector

Anexo D Clasificación de las anomalías

Motivación

El motivo de la implementación de una clasificación de anomalías es mejorar la capacidad de distinguir entre las falencias que presenta nuestra instancia de las actividades. Se tuvieron en cuenta los siguientes pasos para esta adaptación del estándar.

- No había implementación previa, por lo tanto no hubo que adaptar una clasificación anterior.
- Se toma como no necesaria la implementación estricta del estándar, por lo tanto se revisan todas las clasificaciones sugeridas independientemente de si son obligatorias u opcionales.
- Se listan todas las categorías y se seleccionan con los siguientes criterios
 - Las anomalías potenciales, ¿pueden ser clasificadas fácilmente por este criterio?
 - La información que proporciona ésta clasificación:
 - ¿Puede resultar útil en un futuro?
 - ¿Aporta a mejorar la experiencia del usuario?
 - ¿Aporta a la gestión del proyecto?
- Para cada categoría seleccionada, se decide si es esencial para los objetivos del proceso. Si lo es, se la marca como obligatoria.
- Se documenta cada clasificación.

Adherencia

Categoría IEEE Std 1044-1993	Nombre original	Obligatorio en el estándar	Implementada	Obligatorio en el proceso
Causa actual	Actual Cause	✓		
Acción correctiva	Corrective Action			
Valor de cliente	Customer value			
Disposición	Disposition	✓	✓	✓
Misión/seguridad	Mission/safety			
Prioridad	Priority			
Usabilidad	Product status		✓	✓
Actividad	Project activity	✓	✓	✓
Costo	Project cost	✓	✓	✓

Fase	Project phase	✓		
Calidad/ Confiabilidad	Project quality/ reliability			
Riesgo	Project risk			
Calendario	Project Schedule	✓	✓	✓
Repetitividad	Repeatability			
Resolución	Resolution	✓	✓	
Severidad	Severity	✓		
Sociedad	Societal			
Fuente	Source	✓		
Causa probable	Suspected cause			
Síntoma	Symptom	✓	✓	
Tipo	Type	✓	✓	

Clasificaciones

Disposición

La clasificación de disposición describe la manera de la cual fue resuelta la anomalía.

Clasificación	Descripción
Cerrada	La anomalía ha sido abordada y está considerada como completa.
Resolución implementada	La resolución de la anomalía ha sido implementada.
No es un problema	La anomalía no era un problema. El prototipo se comportó de la manera esperada.
No está en el alcance del proyecto	La anomalía no podía ser resuelta, o estaba afuera del alcance como fue definido en las especificaciones.
Duplicada	La anomalía es idéntica a otra anomalía reportada. Referenciar a ésta última.
Diferida	La anomalía será diferida a otra iteración del proceso
Combinado con otro problema	La anomalía es la misma, o va a ser resuelta con la misma implementación de la solución de la anomalía referenciada

Actividad

La clasificación de actividad describe en qué fase del proceso fue generada.

Clasificación	Descripción
Objetivos	La anomalía se debe a un defecto en la identificación de las metas de usabilidad
Contexto	La anomalía se generó en la etapa de identificación de los usuarios y las tareas
Diseño	La anomalía se generó en la etapa de soluciones de diseño

Costo

La clasificación de costo describe una estimación aproximada del costo incurrido en abordar el defecto.

Clasificación	Descripción
Alto	Resolver esta anomalía será extremadamente costoso. Este cambio requiere un gran esfuerzo.
Medio	Resolver esta anomalía será costoso. Este cambio requiere un esfuerzo moderado.
Bajo	Resolver esta anomalía tendrá un costo nominal. Este cambio requiere bajo esfuerzo.
Ninguno	Resolver esta anomalía o implementar este cambio está incluido en el presupuesto del proyecto, por lo tanto no hay un costo adicional asociado.

Calendario

La clasificación de calendario describe una estimación aproximada del tiempo requerido para abordar el defecto o mejora.

Clasificación	Descripción
Alto	Resolver esta anomalía implicaría destruir el calendario. Este cambio requiere un gran esfuerzo
Medio	Resolver esta anomalía cambiaría el calendario significativamente. Este cambio requiere un esfuerzo moderado
Bajo	Resolver esta anomalía tendrá poco impacto en el calendario. Este cambio requiere bajo esfuerzo
Ninguno	Resolver esta anomalía o implementar este cambio está planificado en el calendario, por lo tanto no hay impacto en el calendario asociado

Resolución

La clasificación de resolución describe la decisión que se tomó para superar la anomalía reportada.

Clasificación	Descripción
Inmediata	La modificación será implementada en la iteración actual.
Corrección del prototipo	La modificación será a través de un cambio en el diseño del prototipo.
Actualizar documentación	La modificación será a través de un cambio de la documentación.
Capacitación del usuario	El problema se resolverá entrenando al usuario para la operación del sistema.
Eventual	La modificación será implementada en una iteración futura. Especificar la iteración o fecha límite de resolución.
Corrección del prototipo	La modificación será a través de un cambio en el diseño del prototipo.
Actualizar documentación	La modificación será a través de un cambio de la documentación.
Capacitación del usuario	El problema se resolverá entrenando al usuario para la operación del sistema.
Diferida	La modificación será abordada en una iteración futura.
Resolver en otra entrega	La modificación requerida será implementada en una

	versión futura.
Exención requerida	Será muy difícil implementar la resolución de ésta anomalía. Por lo tanto se requiere una exención para que podamos entregar la próxima versión con ella.
No se resuelve	La modificación no va a ser implementada ni en ésta ni en otra iteración.
No es un problema	La anomalía no era un problema. El prototipo se comportó de la manera esperada.
Exención requerida	Será muy difícil implementar la resolución de ésta anomalía, por lo tanto se requiere una exención para que podamos hacer la entrega sin este cambio.
Resolución no justificada	Resolver este problema o mejora no justifica el tiempo o dinero destinado para implementarlo. O, hacer esta modificación traería más problemas de los que resuelve.
Resolución no identificada	No se pudo encontrar una solución apropiada para el problema reportado.
Obsoleta	El problema ha sido superado por la resolución de otra anomalía u otra causa. Justificar la causa.

Síntoma

La clasificación de síntomas describe cuál es la percepción del usuario del error

Clasificación	Descripción
Problema de entrada	El prototipo no maneja el ingreso de datos de la manera que el usuario espera
Entrada correcta no aceptada	El prototipo no acepta el ingreso de datos correctamente
Entrada incorrecta aceptada	El prototipo acepta el ingreso de datos incorrectos que potencialmente van a producir resultados incorrectos. No se muestran mensajes de error.
Descripciones incorrectas o faltantes	El prototipo no proporciona suficiente descripción para orientar al usuario para el ingreso de datos
Parámetros incompletos o faltantes	El prototipo no muestra todas las opciones o parámetros de entrada necesarios
Problema de salida	El prototipo no presenta la salida de datos de la manera que el usuario espera
Formato incorrecto	El prototipo no presenta los datos en el formato correcto
Resultados/datos incorrectos	El prototipo no presenta los datos esperados o no están enmarcados en el contexto
Ortografía/Gramática	El prototipo presenta errores de ortografía o gramática en la presentación
Cosmético	El prototipo presenta imágenes inadecuadas, o no es percibido positivamente por el usuario debido a problemas estéticos
Falla total percibida	El prototipo no cumple con uno o más requerimientos críticos
Otro	Otro síntoma que no se encuadra en cualquier otra categoría

Tipo

La clasificación de tipo describe con más detalle de qué tipo de problema se trata.

Clasificación	Descripción
Problema de documentación	Problema específico en la documentación
Declaración ambigua	La declaración puede ser interpretada de maneras distintas
Ítem incompleto	La declaración o descripción no parece considerar todos los aspectos de la situación que intenta describir
Ítem incorrecto	La declaración o descripción es incorrecta
Ítem faltante	La declaración o descripción que debe ser incluida en el documento está faltando
Ítems conflictivos	Dos o más declaraciones o descripciones están en conflicto o se contradicen
Ítem confuso	La declaración o descripción confunde al lector
Ítem redundante	La declaración repite otra declaración y resta claridad
Ítem ilógico	La declaración no tiene sentido en referencia a otras declaraciones dentro del mismo documento u otros documentos a los que se refieren
Ítem no verificable	La declaración no puede ser verificada por ningún medio factible
Ítem inalcanzable	La declaración no puede ser garantizada en el ciclo de vida del producto
Problema de calidad del documento	Problema con el documento en sí
Inconformidad de estándares	Los estándares del documento en cuestión establecidos por las políticas de la organización no son alcanzados
No trazable	Los ítems no pueden ser referenciados a documentos previos o posteriores
Desactualizado	El documento no está actualizado a la versión de la iteración actual
Incompleto	El documento tiene una sección importante faltante
Inconsistencia	El documento contiene información que es inconsistente con la información en otro documento
Mejora	Una mejora es una sugerencia que cambia los requerimientos de un producto existente para mejorar la usabilidad o funcionalidad del producto
Requerimientos	Agregar una nueva prestación, remover una prestación innecesaria o modificar una prestación existente para satisfacer más necesidades del usuario
Nueva prestación	
Remover prestación innecesaria	
Mejorar usabilidad	Cambiar la especificación de un artefacto para mejorar la experiencia del usuario
Otra	Una mejora distinta de las mencionadas anteriormente
Falla causada por corrección previa	La falla fue causada por la corrección de otro problema o mejora
Problema conformidad de estándar	El producto resultante no cumplirá con algún estándar, donde ésta conformidad está explícitamente especificada en los requerimientos

Otra	No entra en ninguna de las clasificaciones anteriores
------	---

Usabilidad

La clasificación de usabilidad describe de qué manera la usabilidad del producto podría verse afectada.

Clasificación	Descripción
Inutilizable	El producto no podrá utilizarse con ésta anomalía presente
Degradada	Algunos aspectos del producto funcionarían, pero otros atributos no lo hacen en la condición actual
Afectada	El producto puede ser utilizado, pero una solución alternativa (de la del método preferido por el usuario) debe ser utilizada para lograr algunos objetivos
No afectada	La usabilidad del producto no es afectada

Glosario

Anomalía: Cualquier condición que se desvía de lo esperado basado en la especificación de diseño o de las percepciones o experiencias de alguien. Las anomalías pueden ser encontradas en la revisión, prueba, análisis, compilación o el uso de los productos de software o la documentación aplicada [24].

Proceso: un conjunto de actividades que se interrelacionan o interactúan para transformar entradas en salidas [25].

Actividad: una colección de tareas relacionadas [25].

Usabilidad: Atributo cualitativo que evalúa cuán fácil de usar es una interfaz de usuario.

Efectividad: Que el usuario pueda completar su tarea correctamente.

Eficiencia: La rapidez con que pueden completar las tareas.

Satisfacción: Que el usuario pueda percibir que ha realizado su tarea de manera correcta.

Bibliografía

- [1] IEEE Computer Society, *SWEBOK - Guide to the Software Engineering Body of Knowledge.*, 2004.
- [2] Ian Sommerville, *Software engineering*, Octava edición ed., Harlow, Ed.: Addison Wesley, 2007.
- [3] International Organization for Standardization, *ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability*, Primera edición ed. Geneva, Switzerland, 1998.
- [4] Barry Boehm, "A Spiral Model of Software Development and Enhancement," *ACM SIGSOFT Software Engineering Notes*, vol. 11, no. 4, pp. 14-24, Agosto 1986.
- [5] Standish Group, *The CHAOS Report.*, 1994.
- [6] Robert Wysocki, *Project Management Process Improvement.*, 2004.
- [7] Lautaro Mazzitelli, *Perfil de usabilidad para UML: definiendo la interacción con el usuario*. Buenos Aires, Argentina, 2009.
- [8] Maia Naftali, *Análisis e Integración de métricas para la Accesibilidad Web*. Buenos Aires, 2010.
- [9] IEEE, "IEEE Std 830-1998 Recommended Practice for Software Requirements Specifications," 1998.
- [10] Didar Zowghi and Vincenzo Gervasi, "The Three Cs of Requirements: Consistency, Completeness, and Correctness," in *Proceedings of REFSQ'02*, 2002.
- [11] JCSP Leite and APM Franco, "A Strategy for Conceptual Model Acquisition," in *Proceedings of the First IEEE International Symposium on Requirements Engineering*, San Diego, 1994, pp. 243-246.
- [12] JCSP Leite et al., "Enhancing a Requirements Baseline with Scenarios,"

- [13] John Mylopoulos, Lawrence Chung, and Brian Nixon, "Representing and Using Non-Functional Requirements: A Process-Oriented Approach," Universidad de Toronto, Toronto, 1992.
- [14] Jakob Nielsen, *Usability Engineering*.: Academic Press, 1993.
- [15] Alan M. Davis, "Operational Prototyping: A new Development Approach," *IEEE Software*, p. 71, Septiembre 1992.
- [16] Dan M. Brown, *Communicating Design: Developing Web Site Documentation for Design and Planning*, Segunda edición ed.: New Riders, 2011, p. 169.
- [17] Jesse James Garrett, *The elements of user experience*, Segunda Edición ed.: New Riders, 2011.
- [18] T. R. G. Green, "Instructions and Descriptions: some cognitive aspects of programming and similar activities," 2000.
- [19] International Organization for Standardization, "ISO 13407 - Human-centered design processes for interactive systems," Geneva, 1999.
- [20] Whitney Quesenbery, "Dimensions of Usability: Defining the Conversation, Driving the Process," in *UPA 2003 Conference*, 2003.
- [21] Soren Lauesen and Houman Younessi, "Six styles for usability requirements," in *Proceedings of REFSQ'98*, 1988.
- [22] Martin C Maguire, *RESPECT User-Centred Requirements Handbook*., 1998.
- [23] Russ Unger and Carolyn Chandler, *A Project Guide to UX Design: For user experience designers in the field or in the making*., 2009.
- [24] IEEE, *IEEE 1044.1-1995 Guide to Classification for Software Anomalies*., 1995.
- [25] International Organization for Standardization, *ISO/IEC 90003 Software Engineering - Guidelines for the application of ISO 9001:2000 to computer software*. Geneva, Switzerland, 2004.