

Universidad de las Ciencias Informáticas



Título: Procedimiento para incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software.

Tesis en opción al título de
Máster en Calidad de Software

Autora: Ing. Yanetsi Millet Lombida

Tutora: Dr. C. Ailyn Febles Estrada, Lic.

MSc. Yamilis Fernández Pérez, Ing.

La Habana, 2015.

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del presente trabajo. Autorizo al Centro Nacional de Calidad de Software (CALISOFT) y a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____
__ del año _____.

Ing. Yanetsi Millet Lombida
Autora

Dr.C. Aylin Febles Estrada
Tutora

MSc. Yamilis Fernández Pérez
Tutora

DEDICATORIA

*A mis padres,
para quienes no encuentro palabras para expresar todo lo que siento por ellos,
por enseñarme a ser mejor persona cada día, y poder contar con ellos siempre que los
necesito.*

*A mi hermano,
por ser un ejemplo de consagración para mí en todo momento
y hacerme saber que la tesis ya era hora de terminarla.*

*A mi hija Vanessita,
por ser la alegría más bella que me pudo dar la vida
y expresarme tanto cariño, amor y sonrisas.*

*A mi esposo,
pues sin su apoyo, paciencia y amor esta investigación no habría sido posible.*

AGRADECIMIENTOS

A mi tutora Yamilis por haberme guiado con sus ideas brillantes, ser autora a la par de la investigación y dedicar parte de su poco tiempo para culminar con éxito.

A mi tutora Ailyn por aceptarme como maestrante, por sus valiosos comentarios y sugerencias en el desarrollo de la investigación.

A Yisel y Grether, por darme la oportunidad de conocerlas y contar con ellas cada vez que lo requerí.

A Iliana, por sus iniciativas en la elaboración del documento e ideas que ayudaron a tomar decisiones.

A mis compañeros del grupo de usabilidad, en especial a Adis, Delmys, Arianna, Dimelza y Yenier por nunca escatimar esfuerzos para apoyarme y ayudarme.

A Delvis, Lisandra por sus ayudas en el período que más lo necesité, y colaboración brindada en el desarrollo de esta investigación.

A los que alguna vez me preguntaron ¿cómo va la tesis?, ¿en qué te podemos ayudar?...

A todos, muchas gracias de corazón.

RESUMEN

La usabilidad constituye un factor esencial en el desarrollo de sistemas informáticos, propicia productos atractivos, operables, fácil de comprender y aprender a usar, lo que ayuda a incrementar la cantidad de usuarios y su satisfacción. El presente trabajo se centra en desarrollar un procedimiento que permite incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software, evitando el sobre diseño y elevados cambios posteriores en el producto. Realizar una correcta Ingeniería de Usabilidad desde etapas tempranas de su desarrollo implica que en el proceso de evaluaciones disminuya la cantidad de No Conformidades de usabilidad. Para lograr este propósito se definen requisitos y patrones de usabilidad acorde al modelo de calidad propuesto por la ISO/IEC 25010:2011, así como roles y responsabilidades necesarios a utilizar en los proyectos productivos. La investigación se valida a través de la triangulación de métodos cualitativos y cuantitativos que evidencian el cumplimiento del objetivo trazado.

Palabras clave: usabilidad, Ingeniería de Usabilidad, requisitos, patrones.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO I: MARCO TEÓRICO DE LA INGENIERÍA DE LA USABILIDAD EN EL PROCESO DE DESARROLLO DEL SOFTWARE	10
1.1 Calidad de software	10
1.2 Usabilidad	11
1.3 Ingeniería de Usabilidad (IU).....	17
1.4 La Ingeniería de Usabilidad a nivel de requisitos.....	18
1.4.1 Gestión de los requisitos de usabilidad	20
1.4.2 Técnicas para desarrollar requisitos de usabilidad	21
1.4.3 Metodologías web y herramientas que evalúan los requisitos de usabilidad	26
1.5 La Ingeniería de Usabilidad en la arquitectura de software. Patrones de usabilidad	32
1.6 Análisis de la usabilidad en los proyectos de la UCI.....	33
CAPÍTULO II: PROCEDIMIENTO PARA INCORPORAR LA INGENIERÍA DE USABILIDAD DESDE ETAPAS TEMPRANAS	37
2.1 Diagnóstico inicial	37
2.2 Propuesta del procedimiento.....	38
2.3 Descripción gráfica y textual de la incorporación de la Ingeniería de Usabilidad a nivel de requisitos	39
2.3.1 Técnicas y herramienta propuestas para el desarrollo de requisitos de usabilidad	44
2.3.2 Propuesta de requisitos de usabilidad	49
2.4 Descripción gráfica y textual de la incorporación de la Ingeniería de Usabilidad a nivel de arquitectura de software.....	52
2.4.1 Incorporación de la Ingeniería de Usabilidad en la arquitectura de software. Patrones de usabilidad y su relación con los atributos y propiedades de usabilidad	55
2.5 Roles y responsabilidades que intervienen en la Ingeniería de Usabilidad.....	57
CAPÍTULO III: VALIDACIÓN DEL PROCEDIMIENTO.....	61
3.1 Consulta a expertos	61
3.1.1 Selección de expertos.....	61
3.1.2 Escalamiento de Likert.....	63
3.2 Técnica de ladov	64
3.3 Grupos focales	67
3.4 Entrevista a profundidad	68
3.5 Cuasi experimento con postprueba únicamente y grupos intactos para medir la variable nivel de disminución de la cantidad de NC de usabilidad	69

ÍNDICE DE CONTENIDO

3.6 Triangulación metodológica	76
CONCLUSIONES.....	79
RECOMENDACIONES.....	80
REFERENCIAS BIBLIOGRÁFICAS.....	81
BIBLIOGRAFÍA CONSULTADA.....	85
ANEXOS	89

ÍNDICE DE FIGURAS

Figura 1: Estructura del Departamento de Evaluación de Productos.	4
Figura 2: Cantidad de NC de usabilidad respecto a NC totales en el período 2012-2013.	5
Figura 3: Cantidad de NC de usabilidad respecto a NC totales en el 2014.	6
Figura 4: Etapas del desarrollo de requisitos.	20
Figura 5: Resultados de investigaciones respecto a guías de usabilidad.	21
Figura 6: Metodologías web que consideran a la usabilidad en el desarrollo del software.	26
Figura 7: Clasificación de las NC de usabilidad detectadas en el período 2012-2013.	34
Figura 8: Clasificación de las NC de usabilidad detectadas en el 2014.	35
Figura 9: Etapas de la Ingeniería de Usabilidad insertada en el proceso de desarrollo del software.	39
Figura 10: Descripción gráfica del desarrollo de los requisitos de usabilidad.	41
Figura 11: Descripción gráfica de la arquitectura de software.	53
Figura 12: Relación entre atributos, propiedades y patrones de usabilidad.	56
Figura 13: Roles involucrados en el proceso de desarrollo de la usabilidad.	60
Figura 14: Resultados de la aplicación de la escala de Likert.	64
Figura 15: Resultado de la aplicación de la técnica de ladov y valor del Índice de Satisfacción Grupal (ISG).	66
Figura 16: Métodos de validación de la propuesta.	77

ÍNDICE DE TABLAS

Tabla 1: Características fundamentales para medir la usabilidad.	16
Tabla 2: Resumen de técnicas para desarrollar requisitos.	25
Tabla 3: Metodologías web que evalúan los requisitos de usabilidad.	32
Tabla 4: Técnicas propuestas para el desarrollo de los requisitos de usabilidad.	45
Tabla 5: Estructura de la lista de chequeo.	46
Tabla 6: Indicadores incorporados a la lista de chequeo de requisitos del DEPSW de CALISOFT.	48
Tabla 7: Escala de puntuación para calcular coeficiente de argumentación.	62
Tabla 8: Grado y coeficiente de competencia, coeficiente de conocimiento y argumentación de los expertos.....	63
Tabla 9: Cuadro Lógico de ladov.	65
Tabla 10: Descripción de NC de un producto de software antes y después de la propuesta.	70
Tabla 11: Cantidad de NC de productos antes y después de la propuesta.	76

INTRODUCCIÓN

A nivel mundial, uno de los pilares fundamentales de la informática es el desarrollo del software, actividad que posee un impacto directo en todos los ámbitos de la sociedad moderna, a pesar de las posibles dificultades que pueden surgir en el transcurso del desarrollo, entre ellos: inconsistencia en su funcionamiento, retrasos en su construcción y sobre todo, falta de calidad. El desarrollo del software ha aumentado considerablemente y cada vez son más los sectores de la sociedad que automatizan sus procesos (la medicina, la aeronáutica, la educación, la economía, etc.), de ahí que la calidad en el software sea un elemento imprescindible, por su repercusión en los costes finales, como elemento diferenciador de la competencia y de imagen frente a sus clientes (Brito Riverol 2014). En este sentido, cabe destacar el informe de la *CHAOS Reports* del *Standish Group* en el 2012 (Ver Anexo 1), el cual señala que solo el 39% de los proyectos informáticos finalizan satisfactoriamente, con los recursos planificados y con una calidad aceptable, mientras que el 43% no llegan a finalizar nunca (Group 2013). La buena calidad de los desarrollos y la conciencia colectiva en el sector empresarial de la importancia de las áreas de pruebas, han contribuido a que el mercado internacional proponga nuevos estándares de aceptación (Cusmai 2011).

En la actualidad, las empresas que producen software tienen que velar necesariamente por la calidad de su producción, debido a la competitividad que existe en esta rama y a la velocidad de respuesta que es necesario brindar para aumentar la satisfacción del cliente (Guerrero Llerena 2004). Muchos de los desarrolladores de software piensan que la calidad solo es aplicable al producto, esta se debe medir desde su inicio hasta la entrega al usuario. En el concepto de calidad de un sistema, la usabilidad es una característica clave en el desarrollo del software, lo que permite medir la facilidad de uso, aprendizaje, así como la flexibilidad necesaria para atraer a los usuarios.

Atendiendo a lo expresado anteriormente, se puede decir que enriquecer el proceso de desarrollo del software a partir de la introducción temprana de requisitos de usabilidad en un producto es viable, pues se brinda una clara visibilidad de los aspectos de usabilidad para los desarrolladores y para los usuarios de pruebas, de esta manera se evita impactar negativamente en la duración y costos del proyecto (Florián, Solarte et al. 2010). Las valoraciones de usabilidad en software se han realizado desde 1971. Para el año 2003 el *National Institute of Standards and Technology (NIST)* identificaba más de 30 técnicas para conducir valoraciones de usabilidad (Ramli and Jaafar 2008).

La usabilidad es un tema crítico para la aceptación de un sistema. Si el sistema informático no es percibido como una herramienta que ayuda al usuario a realizar sus tareas, se dificulta la aceptación del mismo. Puede ocurrir que el sistema no llegue a usarse en absoluto, o que se use con escasa eficiencia. Si las tareas del usuario no son respaldadas convenientemente por el

sistema, entonces no se está respondiendo adecuadamente a las necesidades del usuario, y el equipo de desarrollo se está alejando del objetivo principal de la construcción de un sistema software (Ferré Grau 2010), (Mascheroni, Greiner et al. 2012).

Enfocando la usabilidad en las organizaciones desarrolladoras de software, se puede decir que las mismas están empezando a incluir requisitos de usabilidad en las especificaciones de requisitos de software, ya que se han percatado de la importancia que se le está dando en la actualidad a desarrollar productos “usables” y de esta forma atraer mayor cantidad de usuarios. Por otra parte, en los desarrollos de software, la competitividad actual obliga a mejorar el nivel de usabilidad, porque existe un riesgo real de perder clientes si la aplicación no es fácil de usar y/o fácil de aprender (Ferré 2003). Muchas son las ventajas que esta puede proporcionar, y por ello, debería ser tratada como un elemento de calidad estratégico y relevante. Entre sus principales beneficios se encuentran (Pressman and Joyanes Aguilar 1997):

- Reducción de los costes de aprendizaje.
- Disminución de los costes de asistencia y ayuda al usuario.
- Disminución en la tasa de errores cometidos por el usuario y del retrabajo.
- Optimización de los costes de diseño, rediseño y mantenimiento.
- Aumento de la tasa de conversión de visitantes a clientes de un sitio web.
- Aumento de la satisfacción y comodidad del usuario.
- Mejora la imagen y el prestigio de la organización.
- Mejora la calidad de vida de los usuarios, ya que reduce su estrés, incrementa la satisfacción y la productividad.

Estos beneficios implican una reducción de los costes y tiempo de producción, los cuales pueden ser reducidos evitando el sobre diseño y el número de cambios posteriores en el producto. Los sistemas informáticos que son fáciles de usar requieren menos entrenamiento y soporte para el usuario, mientras que los difíciles de usar, disminuyen la salud, bienestar y pueden incrementar la falta de motivación. Tales sistemas suponen pérdidas de tiempo y no son explotados en su totalidad, debido a que el usuario pierde interés en el uso de las características avanzadas del producto y en algunos casos no los utiliza nunca (Lorés, Sendín et al. 2001).

Si bien la usabilidad es un factor importante de calidad, esta no es considerada en etapas tempranas del desarrollo del software, sino más bien cuando el producto ya está casi terminado con sus interfaces listas. Por este motivo, sería interesante poder incluir aspectos concernientes a la usabilidad durante el proceso de captura de requisitos (Ormeño, Panach et al. 2012).

Desde un enfoque del diseño y evaluación de aplicaciones de software, se habla de la usabilidad de software como un área incluida en el campo de la Interacción Persona Ordenador (IPO), que se define como un conjunto de fundamentos teóricos y metodológicos que aseguran el cumplimiento de los niveles de usabilidad requeridos. Se trata, fundamentalmente, de decidir qué

atributos del concepto de usabilidad deben ser priorizados para lograr metas verificables y medibles, cuyo objetivo último, es atraer la atención del usuario lo más posible, ya sea por motivos económicos, publicitarios u otros.

La correcta Ingeniería de Usabilidad conduce a un software usable, productivo y clientes satisfechos (Folmer and Bosch 2004). La opinión más extendida en el campo de la ingeniería de software es que la usabilidad es principalmente relacionada con la interfaz de usuario ((Seffah and Metzker 2004), (Folmer and Bosch 2004)). En otras palabras, la usabilidad afecta principalmente a la interfaz de usuario y no al núcleo del sistema. Por lo tanto, la buena práctica de diseño que separa la interfaz de usuario de la funcionalidad básica, puede ser suficiente para apoyar el desarrollo de un sistema de software usable (Juristo, Moreno et al. 2007).

La autora de la presente investigación considera que el diseño de la interfaz de usuario es un proceso más complejo y constituye un elemento importante a la hora de elaborar un producto, al tener en cuenta una mejor navegabilidad y funcionamiento. Este proceso no debe realizarse a la ligera ya que facilita el trabajo al cliente con el sistema, problema que está dado por falta de roles en los proyectos como el diseñador y el arquitecto de información. La usabilidad es más que la relación con la interfaz de usuario, es la parte visible del sistema (botones, menús desplegables, fondo, color, etc.) y se ocupa de toda la interacción usuario-sistema. La interacción es un concepto más amplio, se encuentra vinculada a la coordinación de intercambio de información entre el usuario y el sistema (Juristo, Moreno et al. 2007).

La Universidad de las Ciencias Informáticas (UCI) a partir del año 2004, comienza a tener una presencia productiva en la industria del software con el desarrollo de proyectos informáticos, fundamentalmente relacionados con la informatización de la sociedad y la exportación; por tal motivo, más del 70% de los estudiantes se encuentran incorporados a proyectos productivos e investigativos de software (Hernández Calzada 2013). Dentro de la infraestructura productiva de la universidad estaba la necesidad de medir la calidad del software, es por eso que surge el Centro Nacional de Calidad de Software (CALISOFT), adscrito al Ministerio de Comunicaciones (MINCOM). Se constituye como organización enfocada a contribuir al desarrollo de la industria, proponiendo al Ministerio normativas técnicas, políticas, procedimientos y estándares para las entidades que desarrollan aplicaciones informáticas. CALISOFT tiene como objeto social además, la responsabilidad de evaluar la conformidad con las normas y emitir certificación a productos informáticos nacionales o importados; brindar servicios de consultorías asociadas a la ingeniería y calidad de aplicaciones informáticas; controlar y auditar el uso de normativas técnicas, procedimientos, documentos estandarizados y buenas prácticas para el desarrollo de aplicaciones y sistemas informáticos en el país. También brinda servicios de formación en los temas de calidad e ingeniería de aplicaciones informáticas (Ministerio de economía y planificación 2012).

Dicho centro en su estructura tiene el Departamento de Evaluación de Productos (DEPSW) y este a su vez cuenta con tres subgrupos: Grupo de Ingeniería de Pruebas de Software (GIPS), Grupo de Seguimiento y Control de la Calidad de los Servicios (SCCS) y un Laboratorio Industrial de Pruebas de Software (LIPS), como parte esencial del DEPSW, cuyo objetivo fundamental es la realización de pruebas de liberación con la calidad requerida a los proyectos productivos de la universidad, así como a los solicitados en el resto del país (Ver Figura 1). El DEPSW evalúa la calidad del software a través de la NC ISO/IEC 9126-1 2005. Una de la característica a medir de esta norma es la usabilidad, la cual es afectada al tenerse presente cuando el software está terminado. Si no se realiza una correcta evaluación de la usabilidad se habrán gastado recursos y tiempo innecesariamente en el desarrollo del producto.

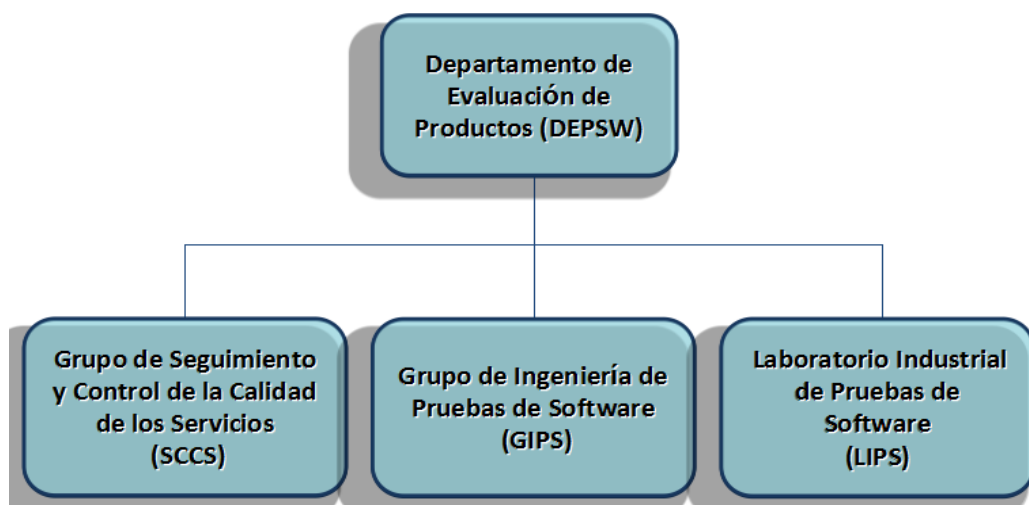


Figura 1: Estructura del Departamento de Evaluación de Productos.

Fuente: Elaboración propia.

Las evaluaciones de usabilidad en la UCI se comenzaron a registrar en el 2012. En el período comprendido de septiembre de ese año a diciembre del 2013, se realizaron 44 evaluaciones de usabilidad. Se tomó como muestra los resultados obtenidos de 10 proyectos (1, 2, 3, 4, 5, 6, 7, 8, 9 y 10) y se detectó en la primera iteración 8, 46, 41, 21, 63, 96, 85, 103, 68 y 66 No Conformidades (NC) respectivamente en el proceso de evaluación, de ellas 7, 30, 10, 15, 42, 43, 30, 33, 16 y 19 NC de usabilidad (Ver Figura 2).

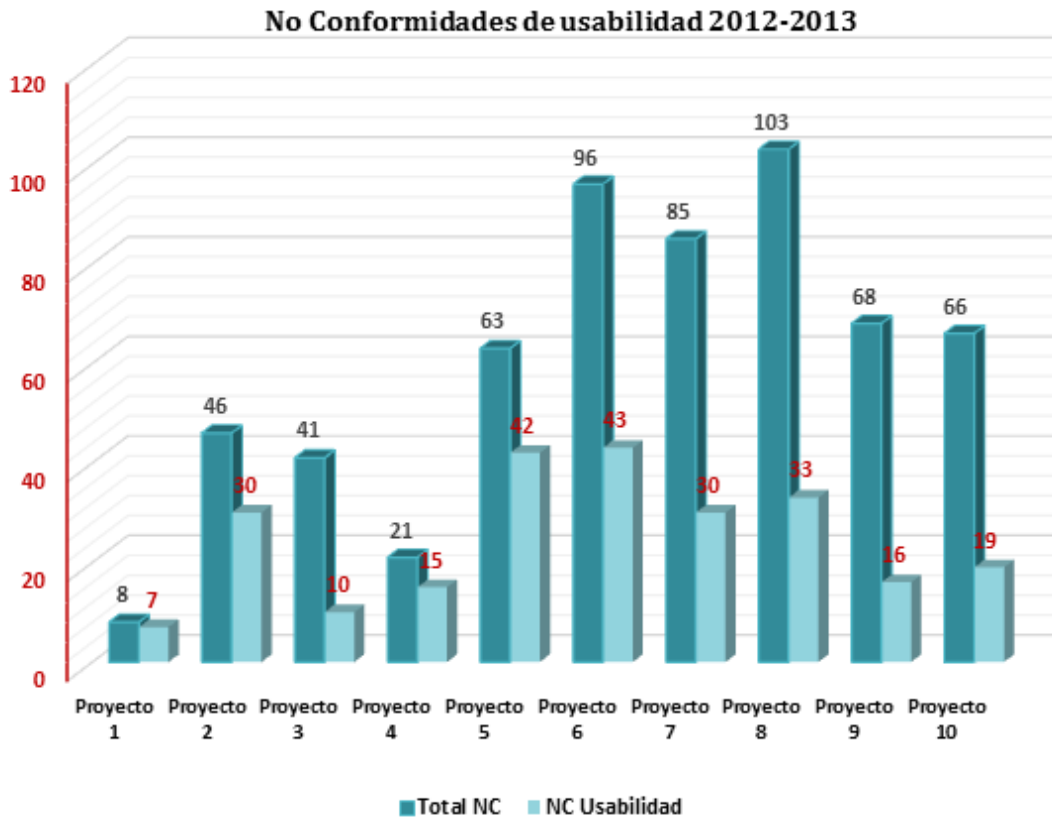


Figura 2: Cantidad de NC de usabilidad respecto a NC totales en el período 2012-2013.

Fuente: Elaboración propia.

En el 2014, maduró la ejecución de las evaluaciones de usabilidad. De las 25 realizadas, se tomó como muestra 8 proyectos (11, 12, 13, 14, 15, 16, 17 y 18) donde se detectó en la primera iteración del proceso de las evaluaciones 79, 14, 9, 101, 252, 36, 71 y 48 NC, siendo de usabilidad 34, 5, 2, 27, 43, 12, 18 y 27 NC (Ver Figura 3).

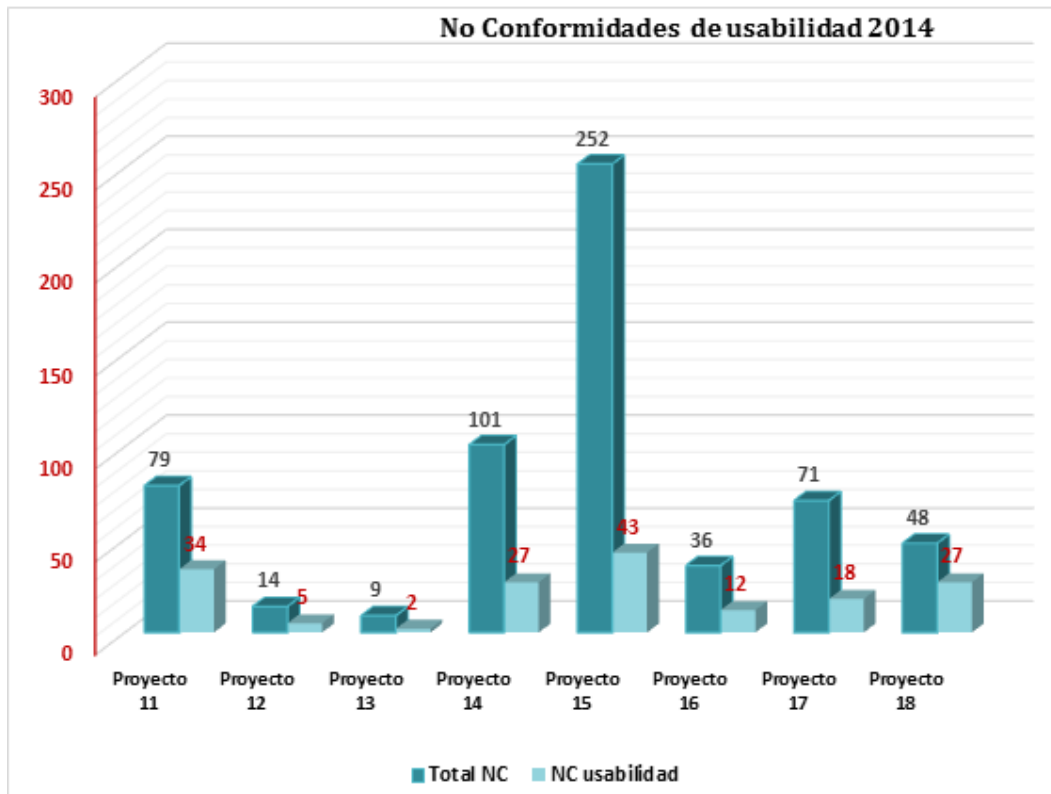


Figura 3: Cantidad de NC de usabilidad respecto a NC totales en el 2014.

Fuente: Elaboración propia.

De esta manera, se evidencia el elevado número de NC de usabilidad detectadas en el proceso de evaluaciones. Las evaluaciones de usabilidad en los productos solicitados a evaluar al DEPSW, trae consigo insuficiencias en los resultados del proceso de evaluación, debido, entre otros aspectos, a la no incorporación de requisitos de usabilidad y patrones desde el inicio del desarrollo de los productos que ayuden a mejorar la usabilidad en los mismos. Aspectos notables al evaluar la satisfacción del usuario, por no familiarizarse con la interfaz del producto, encontrar errores en el mismo o no saber qué hacer en la aplicación.

A partir de este análisis se realizó una entrevista a profundidad a los analistas de software, desarrolladores y administradores de la calidad de algunos proyectos que solicitaron evaluaciones de usabilidad. De acuerdo a sus experiencias en temas de usabilidad, ellos evidenciaron otras dificultades como:

- No se gestionan correctamente los requisitos de usabilidad en los proyectos productivos.
- Presentan limitado conocimiento de la subcaracterística usabilidad de la calidad de software, debido a la inexistencia de una guía específica que oriente al analista cómo gestionar los requisitos de usabilidad.

- Los entrevistados destacan que el porcentaje de código que se dedica al desarrollo de la interfaz, ha aumentado a lo largo ciclo de desarrollo de los productos hasta un promedio de 47-60% del conjunto de la aplicación.
- Obvian características claves diseñadas para hacer más atractivo el producto, bien por desconocer la existencia de patrones de usabilidad, no saber cómo utilizarlos, o no entenderlos.
- Más del 60 % de los proyectos de desarrollo de software sobrepasan su presupuesto, debido a las frecuentes peticiones de cambios en la interfaz de usuario.

Por todo lo antes planteado, se define el siguiente **problema científico**: ¿cómo contribuir a disminuir la cantidad de NC de usabilidad en los productos de software que se evalúan en el Departamento de Evaluación de Productos de CALISOFT?

Se traza el **objeto de estudio** como: proceso de desarrollo de los productos de software y su **campo de acción** se enmarca en: la Ingeniería de Usabilidad en el proceso de desarrollo de un producto de software.

Se define como **objetivo general** desarrollar un procedimiento que incorpore la Ingeniería de Usabilidad en el proceso de desarrollo de un producto de software para disminuir la cantidad de NC de usabilidad en el proceso de evaluaciones en el Departamento de Evaluación de Producto de CALISOFT.

De esta investigación se derivan los **objetivos específicos**:

1. Construir los referentes teóricos relacionando los aspectos fundamentales que sustentan la investigación, mediante los cuales se consulta, extrae y recopila la información relevante sobre la Ingeniería de Usabilidad durante el proceso de desarrollo del software.
2. Diagnosticar el estado que presentan los proyectos productivos en Cuba respecto a la usabilidad en los productos de software.
3. Diseñar un procedimiento para incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software.
4. Validar la solución a través de los métodos: consulta de expertos, grupo focal, técnica de ladov, entrevista a profundidad y cuasi experimento con postprueba únicamente y grupos intactos.

Para resolver el problema de la investigación se propone la siguiente **Hipótesis**: Si se aplica un procedimiento que incorpore la Ingeniería de Usabilidad en el proceso de desarrollo del software entonces disminuirá la cantidad de NC de usabilidad en las evaluaciones realizadas en el Departamento de Evaluación de Productos de CALISOFT.

Definición conceptual de la variable independiente: nivel de incorporación de la Ingeniería de Usabilidad en el proceso de desarrollo del software: capacidad del procedimiento de incorporar

actividades, artefactos, roles y responsabilidades relacionados con el desarrollo de requisitos de usabilidad y diseño de la arquitectura basada en patrones de usabilidad.

Esta variable es discreta y puede asumir uno de los siguientes tres valores:

Nivel 3 (Alto): el procedimiento incorpora actividades en el flujo de trabajo que producen artefactos, integrando la Ingeniería de Usabilidad durante el proceso de desarrollo del software.

Nivel 2 (Medio): el procedimiento incorpora tareas relacionadas con elementos de usabilidad a nivel de requisitos y diseño de interfaz.

Nivel 1 (Bajo): el procedimiento incorpora elementos de usabilidad a nivel de diseño de interfaz.

Definición conceptual de la variable dependiente: nivel de disminución de la cantidad de NC de usabilidad.

Nivel 3 (Alto): cuando la cantidad de NC de usabilidad disminuyen entre 50 % y 100 %.

Nivel 2 (Medio): cuando la cantidad de NC de usabilidad disminuyen entre 30 % y 49 %.

Nivel 1 (Bajo): cuando la cantidad de NC de usabilidad disminuyen menos de un 30 %.

En el desarrollo del trabajo se aplican diferentes métodos científicos: los teóricos y los empíricos.

➤ Métodos teóricos

- Histórico-Lógico, para estudiar la evolución y desarrollo de las etapas de la Ingeniería de Usabilidad de los productos de software desde sus inicios.
- Analítico-Sintético, para analizar las teorías e influencias en el desarrollo de la usabilidad, permitiendo la extracción de los aspectos más importantes para incorporarlos en el proceso de desarrollo del software. Además, se emplean métodos de búsquedas y de consultas bibliográficas.
- Hipotético-Deductivo, para elaborar la hipótesis central de la investigación y ubicar nuevas líneas de trabajo a partir de los resultados parciales.

➤ Métodos empíricos

- La encuesta, se aplica a especialistas del Departamento de Evaluación de Productos Software de CALISOFT, así como a usuarios de diversos proyectos productivos para obtener información, comprender y precisar detalladamente la situación problemática y las deficiencias de usabilidad existentes en la universidad. Así como se le aplica a expertos para diagnosticar y validar la solución que se propone.
- Entrevista a profundidad, con el fin de que especialistas en la temática abordada aporten ideas innovadoras, siempre con el objetivo de lograr una propuesta lo más acertada posible. Además, para la validación de la propuesta como un método cualitativo para tener en cuenta en la triangulación metodológica.
- Triangulación metodológica, con el objetivo de confirmar los resultados obtenidos en las entrevistas a profundidad, grupo focal, técnica de ladov, consulta de expertos y cuasi experimento con postprueba únicamente y grupos intactos.

Resultados a alcanzar:

Como resultado de la presente investigación se obtendrá un procedimiento que incorpore la Ingeniería de la Usabilidad en el proceso de desarrollo de un producto de software, permitiendo organizar y priorizar sus actividades. Además, con la implantación de la Ingeniería de Usabilidad desde la etapa de inicialización del producto, disminuirá la cantidad de NC de usabilidad en las evaluaciones realizadas en el Departamento de Evaluación de Productos de CALISOFT, al hacer más atractivo los productos finales.

Estructura del documento:

La tesis se encuentra estructurada en tres capítulos. En el Capítulo I se elabora el marco teórico referencial, profundizando en los conceptos más relevantes de la característica usabilidad de la calidad de software. Se realiza un estudio del estado del arte de la Ingeniería de Usabilidad con el fin de lograr una mejor comprensión del problema planteado. En el Capítulo II se desarrolla la solución a implantar en los proyectos productivos. En el Capítulo III se realiza la validación de su implantación mediante los métodos entrevista a profundidad, cuasi experimento con postprueba únicamente y grupos intactos, consulta de expertos, grupos focales y técnica de ladov.

Por último se presentan las conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos, que contienen la documentación probatoria de los aspectos más significativos del proceso de investigación.

CAPÍTULO I: MARCO TEÓRICO DE LA INGENIERÍA DE LA USABILIDAD EN EL PROCESO DE DESARROLLO DEL SOFTWARE

Introducción

En el presente capítulo se precisa un conjunto de elementos para conformar el marco teórico relacionado con los aspectos definidos en el objeto de estudio y en el campo de acción. Se comienza definiendo calidad de software, así como se hace énfasis en la usabilidad como característica de calidad y su forma de evaluar. Se realiza una descripción exhaustiva sobre la Ingeniería de Usabilidad, sus metodologías, técnicas y herramientas que esta aporta en el desarrollo y concepción de los productos software, fundamentando la opinión de la autora de la investigación cuando resulta necesario. Se examinan las organizaciones que gestionan los requisitos y patrones de arquitectura de usabilidad en el mundo y por último se analiza cómo ha evolucionado la usabilidad en la producción de software de la Universidad de las Ciencias Informáticas.

1.1 Calidad de software

El software como producto no tangible, hace difícil medir la calidad. La misma suele medirse después de elaborado el producto, resultando muy costoso si se detectan problemas, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad, como su control durante todas las etapas del proceso de desarrollo del software.

Cuando no existe calidad de software se puede presentar las siguientes situaciones (Capote García 2011):

- Programas que no hacen exactamente lo que se espera.
- Proyectos que no terminan nunca.
- Sistemas informáticos que no se utilizan por la dificultad de su manejo.
- Productos software que son imposibles de mantener cuando desaparece la persona o personas que lo desarrollaron.
- Software poco seguro.

Diferentes autores e instituciones han definido la calidad de software. A continuación se citan los más reconocidos en las bibliografías:

La IEEE en su glosario de términos define la calidad como: el grado mediante el cual un sistema, componente o procesos satisface las necesidades o expectativas de un cliente o usuario (IEEE 1990).

Pressman, plantea que la calidad del software es “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente” (Pressman and Hill 2009).

La ISO/IEC 9126:2005 la define como “la totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas” (ISO/IEC 9126-1 2005). Esta norma reconoce seis factores de calidad y 27 sub-atributos.

La ISO/IEC 25000:2011 describe el modelo de calidad interna, externa y de uso del producto software. Es una norma que nace por las inconsistencias entre ISO/IEC 9126:2005 e ISO/IEC 14598:1998, la cual establece los procesos para realizar la evaluación de la calidad del producto software. Su objetivo es aglutinar bajo una misma familia el modelo de calidad y el proceso de evaluación (Capote García 2011). Precisa para la calidad del producto ocho características, seis de ellas contenidas en la ISO/IEC 9126:2005 y se le adicionan dos más.

En la presente investigación se tendrá en cuenta la definición dada por la ISO/IEC 25000:2011 por la madurez que ha adquirido, ya que la autora considera que es la más completa porque tiene en cuenta las necesidades implícitas que debe tener todo producto de software. Además, establece la concordancia con los requisitos funcionales y no funcionales, el uso de normas, estándares y modelos establecidos - internacionalmente, en aras de elevar el control de la calidad del software. Cada estándar mencionado anteriormente proporciona un modelo de calidad, el cual categoriza la calidad del producto en características, que a su vez son descompuestas en subcaracterísticas, conformando una estructura jerárquica. El modelo de calidad del producto, según la 25010 es descompuesto en 8 características, siendo una de ellas la usabilidad.

1.2 Usabilidad

La usabilidad de un producto de software se define como la facilidad de usar la aplicación por los usuarios para lograr sus objetivos. Este término es ambiguo, ya que involucra aspectos objetivos y subjetivos. Entre los elementos objetivos se pueden citar: facilidad de aprendizaje, operabilidad, eficiencia, accesibilidad, comprensibilidad, capacidad para reconocer su adecuación; y los elementos subjetivos lo comprenden: atractividad o satisfacción del usuario, flexibilidad, conformidad, entre otros.

Se puede decir que la usabilidad está determinada por sus contenidos, pautas de diseño, elementos visuales, etc. Mientras más cercanos estén al usuario, mejor y más acertada es la navegación por la aplicación para él. Lógicamente es imposible crear un producto de software ciento por ciento perfecto y en óptimas condiciones, pues no se puede agradar al mismo tiempo a millones de usuarios, sin embargo, los diseñadores deben tratar de mostrar todos los elementos de una manera clara y concisa, minimizando el número de *clícs* y de *scroll* (Pressman and Joyanes Aguilar 1997).

No solo la tecnología y el aspecto gráfico son factores determinantes para hacer un producto de software llamativo, se debe lograr que el usuario encuentre lo que busca en el menor tiempo posible. El término de usabilidad ha sido descrito por diferentes definiciones con sus modelos de

usabilidad y su uso en investigaciones y proyectos. Cada uno descompone esta característica en diferentes subcaracterísticas.

Shackel, fue uno de los primeros autores en el campo de reconocer la importancia de la Ingeniería de Usabilidad y la relatividad de este concepto. Define la usabilidad de un sistema como: “la capacidad en términos funcionales humanos para ser utilizado fácil y eficazmente por rangos específicos de usuarios, teniendo en cuenta especificada formación y apoyo a los usuarios, para cumplir con el grado especificado de las tareas, dentro del rango especificado de los escenarios” (Shackel 1991). Descompone la usabilidad en las siguientes subcaracterísticas:

- Eficacia: rendimiento en la realización de tareas.
- Facilidad de aprendizaje: grado de aprender a realizar las tareas.
- Flexibilidad: adaptación a la variación en las tareas.
- Actitud: satisfacción del usuario con el sistema.

Jakob Nielsen, considerado el padre de la usabilidad, la define como el atributo de calidad que mide lo fácil de usar que son las interfaces web (Nielsen 1993). Es decir, un producto de software usable es aquel en el que los usuarios pueden interactuar de la forma más fácil, cómoda, segura e inteligentemente posible.

La usabilidad es una cualidad demasiado abstracta como para ser medida directamente. Para poder estudiarla se descompone en los siguientes cinco atributos básicos: facilidad de aprendizaje, eficiencia, recuerdo en el tiempo, tasa de errores, y satisfacción (Nielsen 1993):

- Facilidad de aprendizaje: cuán fácil es aprender la funcionalidad básica del sistema como para ser capaz de realizar correctamente la tarea que desea el usuario. Se mide normalmente por el tiempo empleado con el sistema hasta ser capaz de realizar ciertas tareas en menos de un tiempo dado. Este atributo es muy importante para usuarios principiantes.
- Eficiencia: el número de transacciones por unidad de tiempo que el usuario puede realizar usando el sistema. Cuanto mayor es la usabilidad de un producto, más rápido es el usuario al utilizarlo y el trabajo se realiza con mayor rapidez.
- Recuerdo en el tiempo: para usuarios que no utilizan el sistema regularmente es vital ser capaces de usarlo sin tener que aprender cómo funciona partiendo de cero cada vez que lo haga.
- Tasa de errores: este atributo contribuye de forma negativa en la usabilidad de un sistema. Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea.
- Satisfacción: muestra la impresión subjetiva que el usuario obtiene del sistema.

En este modelo, la usabilidad es “Parte de la utilidad del sistema, la cual es parte de la aceptabilidad práctica y, finalmente parte de la aceptabilidad del sistema” (Nielsen 1993).

La **NC-ISO/IEC 9241-11: 1998**, la define como “el grado en que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” (ISO 9241-11 1998).

De acuerdo con esta norma, se basa en los atributos de usabilidad:

- Efectividad: los recursos gastados en relación con la exactitud y la exhaustividad de los usuarios para lograr objetivos.
- Eficiencia: la exactitud e integridad de los usuarios para lograr objetivos específicos.
- Satisfacción: la comodidad y la aceptabilidad del uso.

Como se puede comprobar, en esta definición se relaciona la usabilidad de un sistema a usuarios, necesidades y condiciones específicas. Por tanto, la usabilidad del sistema no es un atributo inherente al software, no puede especificarse independientemente del entorno de uso y de los usuarios concretos que vayan a utilizar el software.

La **NC-ISO/IEC 9126-1:2005**, está presente en la calidad en uso y en la calidad interna/externa. Considera a la usabilidad como: “capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas (ISO/IEC 9126-1 2005). Esta norma especifica la usabilidad por los siguientes atributos medibles:

- Comprensibilidad: capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.
- Aprendizaje: capacidad del producto del software para permitirle al usuario aprender su aplicación.
- Operabilidad: capacidad del producto del software para permitirle al usuario operarlo y controlarlo.
- Atractivo: capacidad del producto del software de ser atractivo o amigable para el usuario.
- Conformidad: capacidad del producto de software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.

La **ISO/IEC 25010:2011**, refiere el mismo concepto que la ISO/IEC 9241-11: 1998, pero tiene en cuenta otros atributos para medir la usabilidad. Para su estudio la descompone en los siguientes atributos (ISO/IEC 25010 2011):

- Capacidad para reconocer su adecuación: capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Capacidad de aprendizaje: capacidad del producto que permite al usuario aprender su aplicación.

- Capacidad para ser usado: capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario: capacidad del sistema para proteger a los usuarios de hacer errores.
- Estética de la interfaz de usuario: capacidad de la interfaz de usuario de agrandar y satisfacer la interacción con el usuario.
- Accesibilidad: capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

La ISO/IEC 9241-11:1998 se utiliza en investigaciones tales como: ((Liang et al. 2009), (Beyan 1992), (Bevan y Macleod 1994), (Hornbæk 2006)), diferentes investigaciones desarrolladas en el año 2012 como ((Dubey, Mittal, et al. 2012) y (Bhatnagar et al. 2012)) añaden la facilidad de aprendizaje. (Winter et al. 2008) agrega seguridad mientras tanto (Dubey, Gulati et al. 2012) adiciona la comprensibilidad y la seguridad. En (M.-Ángel Sicilia et al. 2003) se adicionan atributos como recuerdo en el tiempo y facilidad de aprendizaje. En (Madhavan y Alagarsamy 2014) se añaden error tolerante y facilidad de aprendizaje.

La ISO/IEC 9126-1:2005 se aplica en ((Dasso y Funes 2012) y (Singh y Dubey 2013)). En ((Li y Man 2008) y (Tamir et al. 2008)) lo hacen teniendo en cuenta solo la comprensión, capacidad de aprendizaje y operabilidad.

Otros criterios para diferentes modelos se definieron en ((Beyan 1992), (Chang et al. 1997), (Chang y Dillon 2006), (Nagpal et al. 2013), (M. Sicilia y García 2003), (Juristo et al. 2007), (Costa et al. 2007), (Santos et al. 2007), (Furer et al. 2013), (Basto Cordero et al. 2013), (Jin y Ji 2009)), que no es más que el conjunto de los modelos anteriores, así como la descripción en detalle de algunas opiniones.

A veces es difícil de medir la facilidad de uso teniendo en cuenta el tipo de producto. Por ejemplo: un sitio web ((Nagpal et al. 2013), (Schenkman y Jönsson 2000), (Montazer y Saremi 2008), (Hub y Zatloukal 2010), (Hub y Zatloukal 2009), (Lamichhane y Meesad 2011), (Nagpal et al. 2013) (Sitar-Taut et al. 2009)). Sistema E-learning ((Siu Keung et al. 2003), (Oztekin et al. 2010)). Comunidad virtual (Preece 2001); o componente de software (Bertoa et al. 2006).

En gran medida los atributos en diversos modelos se superponen, pues utilizan en algunos casos otro nombre para el mismo atributo o existen nombres iguales para diferentes atributos. Los autores tienen distintas opiniones acerca de cuáles son indispensables en el cálculo usabilidad. A partir de este análisis, se confirma lo declarado por Folmer y J. Bosch, el término de la usabilidad es aún ambigua y muchos investigadores pasan mucho tiempo en encontrar la mejor manera de definir la usabilidad, a través de la definición de los atributos que se puede medir (Folmer y Bosch, 2004).

La tendencia en los últimos años es utilizar los estándares ISO 9241 y 9126 o la extensión o adaptación de estas normas con un incremento de la ISO 25010 a partir de la aprobación de la familia ISO 25000. Los valores de los atributos de calidad que se derivan de la usabilidad son heterogéneos porque vienen de razones objetivas (sus atributos se pueden medir o evaluar directamente) y subjetivas (sus atributos se miden de acorde a la experiencia del evaluador en el tema). Tras revisar la bibliografía, es posible concluir que la evaluación de la usabilidad es aún un reto, para el proceso de desarrollo de un producto de software.

En la Tabla 1 se realiza un análisis exhaustivo y comparativo de las distintas definiciones de usabilidad estudiadas en el capítulo.

Tabla 1: Características fundamentales para medir la usabilidad.

Fuente: Elaboración propia.

Autores/Atributos		Shacke (1991)	Nielsen (1993)	9241-11 (1998)	Chang 2006	Sicilia 2003	Mich 9126-1 (2005)	Siu 2003	25010 (2011)	Totales
O B J E T I V O S S U B J E T I V O S	Facilidad de aprendizaje	X	X			X	X		X	5
	Recuerdo en el tiempo		X			X				2
	Comprensibilidad						X	X		2
	Capacidad para reconocer su adecuación								X	1
	Eficacia	X								1
	Tasa de errores		X		X			X	X	4
	Eficiencia		X	X	X	X		X		5
	Efectividad			X		X				2
	Operabilidad						X		X	2
	Accesibilidad						X		X	2
	Consistencia				X			X		2
	Internacionalización							X		1
	Flexibilidad	X								1
	Conformidad						X			1
	Actitud	X								1
Satisfacción o atractivo		X	X	X	X		X	X	X	7
Totales	4	5	3	4	5	2	5	5	6	

Luego de analizar la Tabla 1, se evidencia que los autores que más atributos de usabilidad tienen presente son: Nielsen, Sicilia que utiliza la ISO/IEC 9241-11:1998 y agrega dos atributos más, Siu, la ISO/IEC 9126-1:2005 y la ISO/IEC 25010:2011. Esta última es la que se asume en la investigación, por ser una de las más completas en cuanto a la cantidad de atributos y contener los que más consideran los distintos autores evaluar en la usabilidad, como son: facilidad de aprendizaje, estética de la interfaz de usuario, protección contra errores de usuario, capacidad para ser usado. Además, considera la accesibilidad, la cual es muy importante para medir la

evaluación de usabilidad en los productos de software utilizados por usuarios con determinadas características y discapacidades.

Luego de este análisis, la autora de la investigación considera que a partir que una institución se apropie de una definición de usabilidad y de su modelo se puede con relativa facilidad evaluar la usabilidad de los productos de software y determinar sus deficiencias o NC. Se debe analizar cómo considerar cada sub características de usabilidad dentro del proceso de desarrollo del software, pues es demasiado tarde hacerlo al final. Esto lo permite la Ingeniería de Usabilidad, ya que incorpora técnicas que rigen un adecuado nivel de usabilidad en los sistemas informáticos desarrollados para satisfacer las necesidades de los usuarios.

1.3 Ingeniería de Usabilidad (IU)

El término Ingeniería de Usabilidad se introduce por primera vez en *Digital Equipment Corporation*, para referirse al conjunto de conceptos y técnicas que permiten planificar, realizar y verificar los objetivos de la usabilidad de un sistema (Perurena Cancio and Moráguez Bergues 2013).

Para Granollers es una aproximación metodológica que permite desarrollar aplicaciones interactivas con el parámetro de la facilidad de uso o usabilidad como objetivo preferente (Granollers 2004).

Los ingenieros software han visto tradicionalmente la usabilidad como análoga a otros atributos que se prueban en *test* de aseguramiento de calidad. Esta situación ha llevado a la aplicación de las técnicas de usabilidad muy tarde en el ciclo de desarrollo, cuando los principales problemas de usabilidad son demasiado costosos de arreglar. Puesto que estos son fácilmente detectables en las primeras fases del desarrollo, hay una tendencia a integrar la Ingeniería de Usabilidad en el proceso de desarrollo del software, de una forma más compacta (Mascheroni, Greiner et al. 2012). Dentro de este contexto se puede referenciar que la Ingeniería de Usabilidad se realiza en un proceso que tiene tres fases principales: Especificaciones, Diseño y Evaluación. A continuación, se exponen las actividades o tareas realizar en cada etapa del ciclo mencionado (Mascheroni, Greiner et al. 2012):

- Especificaciones: antes de comenzar con el proyecto, se confecciona una lista de especificaciones de usabilidad pretendiendo plasmar los niveles de usabilidad que interesen alcanzar. Las especificaciones orientarán el proceso de desarrollo continuo, pero para fijarlas resulta necesario reconocer previamente a los usuarios y las tareas que van a realizar con el sistema. Estas son: el análisis o distinción de usuarios, identificación de tareas y especificaciones de usabilidad.
- Diseño: luego de realizar el análisis de tareas y sabiendo cuáles van a ser las soportadas por el sistema, se comienza con el diseño de la interacción del sistema, como una primera aproximación que será evaluada y, posteriormente, mejorada con iteraciones de dicho proceso.

- Evaluación de usabilidad: es un proceso que permite determinar el nivel de usabilidad que el prototipo actual del sistema alcanza, y así poder identificar los defectos de usabilidad que este presenta. Existen dos maneras de realizarla: los *test* de usabilidad y las evaluaciones heurísticas. Generalmente, primero se aplican los *test* de usabilidad, y se complementan con evaluaciones heurísticas.

Estas fases buscan alcanzar un mejor nivel de usabilidad en el software desarrollado. Para ello, primeramente se definen unas especificaciones de usabilidad que van a servir de patrón con el que comparar el nivel de usabilidad del sistema, luego comienza un ciclo diseño-evaluación-rediseño, que finaliza cuando se alcanzan los niveles detallados en las especificaciones de usabilidad.

La principal razón por la cual aplicar la Ingeniería de Usabilidad cuando se desarrolla un sistema software, es la obtención de un sistema que hace al usuario más productivo, aumentando su eficiencia y satisfacción al utilizarlo (Ferré Grau 2010). Es necesario realizar una integración de la Ingeniería de Usabilidad en la parte de planificación y gestión de proyectos software (Ferré Grau 2010).

Luego de los conceptos planteados anteriormente, la autora de la investigación aprecia la importancia de incluir la Ingeniería de Usabilidad como complemento fundamental en el proceso de desarrollo del software especificando niveles de usabilidad. Para dar cumplimiento a lo expresado, es necesaria la correcta definición de los requisitos de usabilidad con el objetivo de lograr mejoras en los productos de software en desarrollo.

1.4 La Ingeniería de Usabilidad a nivel de requisitos

Los requisitos son las necesidades del producto que se debe desarrollar en cualquier proyecto de software. De acuerdo a esto, es importante no perder de vista que un requisito debe ser especificado por escrito como todo contrato o acuerdo entre dos partes; posible de probar o verificar para poder comprobar si se cumplió con él o no; consistente para que no entre en contradicción con otros requisitos, y conciso, o sea, fácil de leer y entender. Además, un requisito deber estar completo, es decir, que proporcione la información suficiente para su comprensión y por último, no debe ser ambiguo para no causarle confusiones al lector (Hernández Aguilar 2009).

La IEEE (del inglés *Institute of Electrical and Electronics Engineers*) publicó que un requisito es: “una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal” (IEEE 2006).

Para el SEI (del inglés *Software Engineering Institute*) los requisitos del producto son definidos como un refinamiento de los requisitos del cliente en el lenguaje de los desarrolladores, transformando los requisitos implícitos en requisitos derivados explícitos. El desarrollador usa los

requisitos del producto para guiar el diseño y la construcción del producto (Software Engineering Institute 2010).

La autora del presente trabajo trata los requisitos como establece el SEI, por considerar importante para el desarrollo de esta investigación los requisitos no funcionales, específicamente los de usabilidad, y coincidir con las buenas prácticas que recomienda utilizar el DEPSW de CALISOFT, en los proyectos productivos que se encuentran en desarrollo.

Los requisitos no funcionales son restricciones en el servicio o función ofrecida por el sistema ((IEEE 2006), (Sommerville 2004)). Estos requisitos especifican el comportamiento del producto (Sommerville 2004).

Los requisitos tienen un rol protagónico en el éxito de un proyecto de software. En este sentido ha surgido el término Ingeniería de Requisitos (IR), el cual puede ser descrito como el proceso de descubrir, analizar, documentar y verificar los requisitos para un sistema (Sommerville 2005). De lo antes expresado, se puede inferir que los principales beneficios de la IR están dados por permitir la gestión de las necesidades del proyecto en forma estructurada. Disminuyen los costos y retrasos del mismo, así como mejorar la calidad del software y comunicación entre equipos logrando un consenso entre clientes y desarrolladores en considerar sus requisitos cuidadosamente y revisarlos dentro del marco del problema.

A partir del estudio realizado en las bibliografías ((Pérez Olmo 2009), (Pressman 2002), (Wieggers 2003), (Herrera 2003), (Ramos Blanco 2013)) la autora coincide y afirma que independientemente de que existen diferentes enfoques de procedimientos, en la IR se tiene un denominador común, que consiste en el desarrollo de requisitos y puede resumirse en las siguientes etapas fundamentales: Elicitación, Análisis, Especificación y Validación (Ver Figura 4). Estas actividades ayudan a reconocer la importancia que tiene para el desarrollo de un proyecto de software, realizar una especificación y administración adecuada de los requisitos de los clientes.

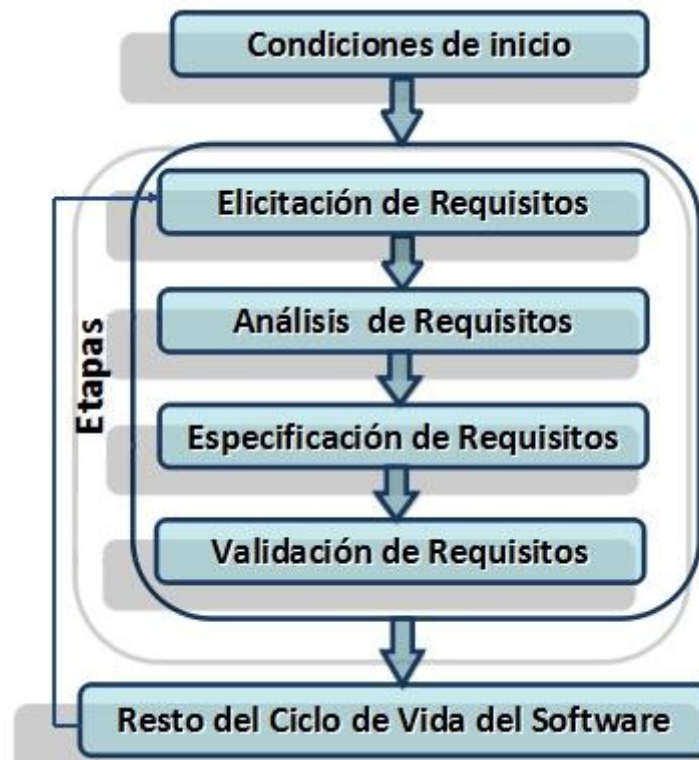


Figura 4: Etapas del desarrollo de requisitos.

Fuente: Elaboración propia.

A la par del desarrollo de la industria del software, ha aumentado el interés por adoptar las mejores prácticas en los requisitos de software. Entre estos requisitos se encuentran los de usabilidad, con el objetivo de incluirse en el proceso de desarrollo del software de un producto.

1.4.1 Gestión de los requisitos de usabilidad

Como parte de la investigación de Ormeño y colectivo de autores se realizó un estudio de casos, basados en la Ingeniería de Usabilidad, con el objetivo de conocer acerca de cómo se trata esta característica. Se puede arribar que el 31.03% de las publicaciones utilizan guías de usabilidad existentes para la elicitación de requisitos de usabilidad. El resultado sobre el uso de guías propuestas (definidas por los mismos autores) es del 24.14%, así como un 44.83% restante hacen referencia a "No existe", es decir que la publicación no indica qué guía usa para este proceso (Ver Figura 5) (Ormeño, Panach et al. 2012).

En las publicaciones seleccionadas, la elicitación de requisitos de usabilidad está comprendida normalmente dentro de la fase del análisis ((Soares and Vrancken 2008), (Escalona and Arag 2008)), es decir, una vez que se hayan elicitado todos los requisitos funcionales. Esta incorporación tardía de los requisitos de usabilidad puede implicar cambios en la arquitectura del sistema, ya que algunos requisitos de usabilidad pueden estar estrechamente ligados a requisitos funcionales ((Bass and John 2003), (Folmer and Bosch 2004)). En el análisis de resultados del estudio sistemático, se observa que la cantidad de publicaciones que abordan de manera clara y

concisa cómo se realiza el proceso de extracción de requisitos de usabilidad en etapas iniciales son mínimas. Por el contrario, se han encontrado muchos trabajos donde la elicitación de requisitos de usabilidad, se realiza en fase de diseño, junto con la elicitación de requisitos de interacción ((Jokela, Seffah et al. 2005), (Sindhgatta and Srinivas 2005), (Jokela, Koivumaa et al. 2006)).

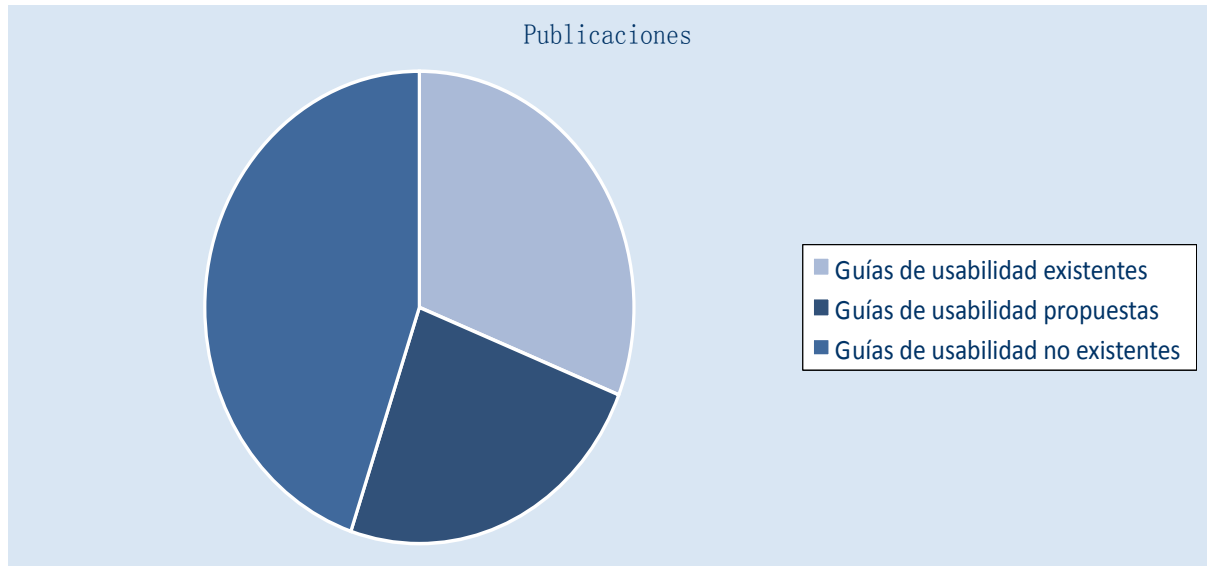


Figura 5: Resultados de investigaciones respecto a guías de usabilidad.

Fuente: Modificado de (Ormeño, Panach et al. 2012).

Los resultados obtenidos de las guías de usabilidad si bien se ajustan al enfoque de la elicitación de requisitos de usabilidad, no solucionan en gran medida los problemas que presentan las aplicaciones en el proceso de desarrollo del software (cambio continuo en los requisitos, presencia de nuevos requisitos, comprensión del mecanismo de interacción con el usuario, aplicación de nuevas guías y estándares para diversas plataformas y tecnologías, entre otros) (Ormeño, Panach et al. 2012).

Las propuestas encontradas en el estudio ayudan a superar en parte el obstáculo de la integración de la Ingeniería de Usabilidad. No obstante, la aplicación de las guías requiere, en general, técnicas para desarrollar requisitos, así como la interpretación de un experto en usabilidad para su correcto uso (Ormeño, Panach et al. 2012).

1.4.2 Técnicas para desarrollar requisitos de usabilidad

Para poder aplicar las técnicas de Ingeniería de Usabilidad en la empresa, la primera tarea a realizar consiste en obtener apoyo de la dirección a los principios de la Ingeniería de Usabilidad, convenciéndoles acerca de los beneficios que se obtienen al aplicarlos (Mascheroni, Greiner et al. 2012).

El éxito en la obtención de requisitos depende del entendimiento a partir de la comunicación entre clientes y desarrolladores. Debido a lo complejidad que esto puede implicar, se ha trabajado

desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa.

Entrevistas

Las entrevistas resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y análisis de los resultados (Pérez Olmo 2009).

La entrevista se define como la conversación de dos o más personas en un lugar determinado para tratar un asunto. Técnicamente es un método de investigación científica que utiliza la comunicación verbal, para recoger informaciones en relación con una determinada finalidad. (López and Deslauriers 2011).

La entrevista no es una técnica sencilla de aplicar, pues requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado (Pérez Olmo 2009).

JAD (*Joint Application Development* / Desarrollo Conjunto de Aplicaciones)

Esta técnica fue introducida por IBM (en inglés *International Business Machines*) a finales de los años 70. Se caracteriza por: realizar reuniones donde participan usuarios claves, gerentes y analistas de sistemas involucrados en el análisis del sistema actual (pretende ser un proceso efectivo), permite determinar conflictos existentes y entender cuáles serán difíciles de resolver, consta de varias sesiones (días y/o semanas) e implica costos de recursos humanos involucrados y de los lugares reservados para realizar los JAD fuera de la organización (Mendoza 2012).

En comparación con las entrevistas individuales, presenta la siguiente ventaja: ahorra tiempo al evitar que las opiniones de los clientes se contrasten por separado (Departamento de Tecnologías de la Información 2010).

Encuestas

Es el método de recogida de información cuantitativa que consiste en interrogar a los miembros de una muestra, sobre la base de un cuestionario estructurado. Además, la encuesta es una técnica de investigación que consiste en una interrogación verbal o escrita que se le realiza a las personas, con el fin de obtener determinada información necesaria para una investigación (Alelú, Cantín et al. 2011). La técnica de encuesta es ampliamente utilizada como procedimiento de investigación, ya que permite obtener y elaborar datos de modo rápido y eficaz. Este procedimiento de investigación posee, entre otras ventajas, la posibilidad de aplicaciones masivas y la obtención de información sobre un amplio abanico de cuestiones a la vez (Casas Anguita, Repullo Labrador et al. 2003).

Tormenta de ideas (*Brainstorming*)

Tormenta de ideas es una técnica de reuniones en grupo, cuyo objetivo es que los participantes muestren sus ideas de forma libre. Las sesiones de *brainstorming* deben estar integradas por un número de cuatro a diez personas, uno de los cuales debe asumir el rol de moderador de la sesión, pero sin carácter de controlador (Pérez Olmo 2009).

Como técnica de captura de requisitos es sencilla de aplicar. Frente al JAD tiene la ventaja de que es muy fácil de aprender y requiere poca organización. Además, suele ofrecer una visión general de las necesidades del sistema, pero normalmente no se obtienen resultados con el mismo nivel de detalle que en otras técnicas. Sus fases son: preparación, generación, consolidación y documentación (Escalona and Koch 2004), (Ramos Blanco 2013)).

Grabaciones de video y de audio

Básicamente existen dos formas de utilizar las grabaciones: como registro y apoyo de las entrevistas, así como para analizar algún proceso en particular. En cuanto a su función de apoyo, es importante porque permite centrar la atención en la entrevista en sí, en vez de distraerse tomando notas de todo lo que se dice. Cuando se trata de analizar algún proceso en particular, su ayuda es inestimable (sobre todo las filmaciones de video) porque permite ver y analizar en detalle ese proceso la cantidad de veces que sea necesario (Davyt Dávila 2001).

Mapas conceptuales

Los mapas conceptuales son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en un lenguaje común. Sin embargo, deben ser usados con cautela porque en algunos casos complejos pueden llegar a ser ambiguos, si no se acompaña de una descripción textual ((Escalona and Koch 2004), (Pérez Olmo 2009)).

Storyboards

Los *storyboards* identifican a los involucrados, explican qué pasa y cómo. Tal vez ninguna técnica de obtención de requisitos ha sido objeto de tantas interpretaciones como la técnica de *storyboard*. Sin embargo, la mayoría de estas interpretaciones coinciden en que el propósito de los *storyboard*, es obtener el criterio de los usuarios sobre las funcionalidades propuestas para la aplicación en etapas tempranas del proceso de desarrollo del software , mucho antes de que sean comprometidos a código ((Leffingwell and Widrig 2003), (Ramos Blanco 2013)).

Aprendiz

Esta técnica se basa en la idea del maestro y en la observación del trabajo real. El ingeniero de software representa al aprendiz y el usuario/cliente cumple el rol de maestro. El aprendiz se sienta con el maestro a aprender por medio de la observación, haciendo preguntas y también realizando

algún trabajo bajo la supervisión del maestro ((Aurum and Wohlin 2005), (Robertson and Robertson 2006), (Pérez Teruel 2007), (Pérez Olmo 2009), (Ramos Blanco 2013)).

Generalmente para el cliente/usuario es difícil explicar completamente su trabajo. En ocasiones tiende a omitir elementos que consideran triviales. La técnica del aprendiz es muy útil para mitigar este problema. Es también una técnica apropiada para un proyecto donde el problema no es estructurado, o sea, en casos en los que se necesita obtener conocimiento tácito de los clientes/usuarios. Tiene como limitante que su implementación requiere de mucho tiempo ((Aurum and Wohlin 2005), (Robertson and Robertson 2006), (Pérez Teruel 2007), (Pérez Olmo 2009), (Ramos Blanco 2013)).

Escenarios de operación

Es posible llevar a cabo de manera informal la obtención de requisitos basada en escenarios cuando los ingenieros de requisitos trabajan con los clientes en la identificación de escenarios y en la captura de detalles de dichos escenarios. Los escenarios se pueden redactar como texto, complementados por diagramas, fotografías de las pantallas, etcétera (Sommerville 2004).

Prototipos

Un prototipo es un borrador de un producto potencial o de una parte del mismo. Es una simulación de los requisitos (INTECO 2008). La elaboración de prototipos de un sistema de información (SI) es una técnica valiosa para la recopilación rápida de información específica acerca de los requisitos de información de los usuarios. Los prototipos efectivos deben hacerse tempranamente en el proceso del desarrollo de sistemas informáticos, es decir, durante la fase de determinación de requisitos. Sin embargo, la elaboración de prototipos es una técnica compleja que requiere el conocimiento completo del proceso de desarrollo del desarrollo de sistemas informáticos antes de que pueda ser lograda satisfactoriamente ((Mendoza 2012), (Ramos Blanco 2013)).

Extracción de fuentes como documentos del negocio, estándares o pautas establecidas en la organización cliente

Otro de los métodos tradicionales para determinar los requisitos es el estudio de los documentos del negocio para descubrir los aspectos importantes, las políticas, las reglas y las directrices, así como ejemplos del uso de la información en la organización ((Duran and Bernárdez 2000), (Mendoza 2012), (Ramos Blanco 2013)). A continuación, en la Tabla 2 se realiza un resumen de las técnicas descritas en este epígrafe.

Tabla 2: Resumen de técnicas para desarrollar requisitos.

Fuente: Elaboración propia.

Técnicas	Ventajas	Desventajas
Entrevista	Visión del problema Fácil de aprender Popular	No es sencilla de aplicar
JAD (Joint Application Development / Desarrollo Conjunto de Aplicaciones)	Ahorra tiempo Visión del problema	Costo de aplicación Demora al aplicarla Difícil de aprender
Encuesta	Popular Fácil de usar Fácil de aprender Precisa Visión del problema	Demora en procesar los datos Interpretación de las preguntas
Tormenta de ideas (Brainstorming)	Sencilla de aplicar Fácil de aprender Bajo costo de aplicación Visión del problema	Poca precisión
Grabaciones de video y de audio	Visión del problema Ayuda inestimable	Costo de aplicación
Mapas conceptuales	Popular Fácil de aprender Visión del problema	Requiere documentación
Storyboards	Visión del problema	Difícil de aprender
Aprendiz	Visión del problema	Demora al aplicarla
Escenarios de operación	Visión del problema	Requiere de conocimiento
Prototipos	Visión del problema Ayuda inestimable	Requiere de conocimiento
Extracción de fuentes como documentos del negocio, estándares o pautas establecidas en la organización cliente	Visión del problema	Requiere documentación

Estas técnicas se utilizan en las distintas fases de la Ingeniería de Requisitos para desarrollar los requisitos con la calidad requerida, de ahí la consideración de aplicarlas en la gestión de los requisitos de usabilidad. Además de las técnicas para enriquecer los requisitos existen metodologías de desarrollo que ayudan a evaluar los requisitos de usabilidad. Por interés de la autora de este trabajo se tratará con mayor profundidad en el próximo epígrafe.

1.4.3 Metodologías web y herramientas que evalúan los requisitos de usabilidad

Varias metodologías de desarrollo web tratan los requerimientos no funcionales, considerando como un atributo relevante la usabilidad. Existen investigaciones que comentan que el requerimiento de usabilidad es considerado por la mayoría de las metodologías, aunque no establecen cómo validar la existencia de este atributo ((Escalona and Koch 2004), (Escalona and González 2007), (Rojo and Oliveros 2012)).

En la actualidad, varias metodologías de desarrollo web tratan los requisitos funcionales y los no funcionales de distinta manera. Estas metodologías son tenidas en cuenta porque el desarrollo de aplicaciones web involucra características que muchas veces no están presentes en el desarrollo de las aplicaciones tradicionales ((Escalona and Koch 2004), (Moreno and Marciszack 2013)). Por otra parte, también intervienen distintos tipos de personas, con diferentes habilidades que contribuyen a la construcción del software, como por ejemplo: diseñadores, analistas, expertos en usabilidad, etc. Todos estos aspectos hacen que se deba trabajar en forma ordenada y coordinada a través de alguna metodología, que establezca de forma clara y precisa los requisitos funcionales y no funcionales, para obtener un sistema informático que cubra todas las expectativas funcionales y de calidad requerida (Moreno and Marciszack 2013).

Las metodologías que se detallan a continuación establecen ambos tipos de requisitos para la construcción de aplicaciones web, entre las cuales se pueden citar: WSDM (Web Site Design Method), HFPM (Hypermedia Flexible Process Modeling), UWE (UML-Based Web Engineering), OOWS (Object Oriented Web Solutions), DDRE (Design-Driven Requirements Elicitation), WebML (Web Modeling Language), NDT (Navigational Development Techniques). Ver Figura 6.

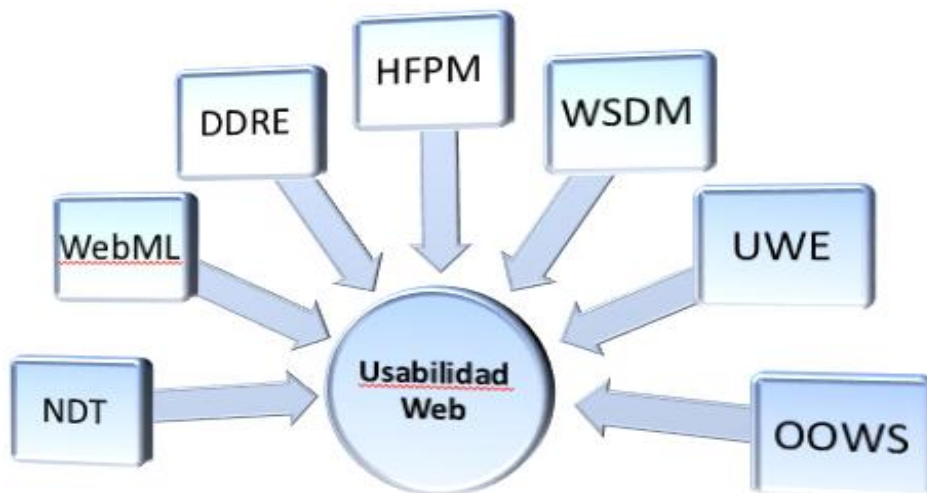


Figura 6: Metodologías web que consideran a la usabilidad en el desarrollo del software.

Fuente: Elaboración propia.

WSDM (*Web Site Design Method*)

Es un Método de Diseño para Sitios Web (*Web Site Design Method*), donde hay un acercamiento al usuario que define los objetos de información, basado en sus requisitos de información para el uso de la Web. Propone cuatro etapas: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. El tratamiento de requisitos se lleva a cabo en la etapa inicial, donde, en primer lugar, se identifican y clasifican los usuarios que van a hacer uso de la aplicación web. A continuación, se describen los requisitos de cada grupo de usuarios y sus fases (Escalona and Koch 2004):

- Fase de Modelo de Usuario, se intenta detectar los perfiles de usuarios para los cuales se construye la aplicación.
- Fase de Diseño Conceptual, se desarrolla durante el modelado conceptual donde se realizan dos tareas a la vez: el modelado de objetos y el diseño de la navegación. Este tipo de diseño de navegación en aplicaciones tiene una estructura muy jerárquica.
- Fase de Diseño de Implementación, se modela la interfaz para cada rol de usuario. Durante esta fase, se tiene presente en el sitio: la construcción de la arquitectura de navegación, la creación y diseño de la página principal disponible, el código de los programas técnicos y la creación de alta funcionalidad teniendo como fin a la animación.
- Fase de Realización de Implementación, se codifican todos estos aspectos en el lenguaje concreto que se haya seleccionado.

Los autores Escalona y González desde el 2007 trabajan en una propuesta de herramienta que permite soportar el proceso de trabajo (Escalona Cuaresma and González Romano 2007).

HFPM (*Hypermedia Flexible Process Modeling*)

Define un proceso detallado que cubre todo el proceso de desarrollo del software y que está compuesto por 13 fases. En la primera de ellas, modelado de requisitos, propone las tareas siguientes:

- Descripción breve del problema.
- Descripción de los requisitos funcionales.
- Realización del modelo de datos.
- Modelado de la interfaz de usuario.
- Modelado de los requisitos no funcionales.

HFPM no está vigente actualmente, sin embargo, fue la primera en definir ciertos aspectos (Escalona and González 2007):

- Incluye al usuario desde el principio del desarrollo.
- Establece la necesidad de definir modelos específicos para el usuario. Aunque no define ninguno.

- Establece la necesidad de elaborar manuales de usuario e incluir esto en el proceso de desarrollo.

Esta metodología en la fase de administración de requisitos no menciona técnicas o herramientas (Rojo 2012) desventaja que percibe la autora de la investigación.

UML-Based Web Engineering (UWE)

La propuesta de Ingeniería Web basada en el Lenguaje de Modelado Unificado (UML) es una metodología detallada para el proceso de autoría de aplicaciones, con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo, puntos de control y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado (Escribano 2010).

Los principales aspectos en los que se fundamenta UWE son los siguientes:

- Uso de una notación estándar UML.
- Definición de métodos de los pasos para la construcción de los diferentes modelos.
- Especificación de restricciones para aumentar la exactitud de los modelos. Se recomienda el uso de restricciones escritas OCL (Lenguaje de restricciones de objetos).

UWE propone una extensión que se utiliza a lo largo del proceso de autoría. Este proceso está dividido en cuatro pasos o actividades:

- Análisis de Requisitos: fija los requisitos de la aplicación web para reflejarlos en un modelo de casos de uso.
- Diseño Conceptual: materializado en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.
- Diseño Navegacional: lo podemos subdividir en: Modelo del Espacio de Navegacional y Modelo de la Estructura de navegación, la cual muestra la forma de navegar ante el espacio de navegación.
- Diseño de Presentación: representa las vistas de la interfaz del usuario mediante modelos estándares de interacción UML.

Contiene herramientas de soporte para el diseño sistemático de aplicaciones web (ArgoUWE y MagicUWE) (Escribano 2010). ArgoUWE es una extensión de la herramienta de código abierto ArgoUML, a la cual se le han añadido capacidad de modelado, navegación y estructuras de presentación.

OOWS (Object Oriented Web Solutions)

Es una extensión del Método Orientado a Objeto (*OO-Method*) basado en modelos, al cual se le añaden a las técnicas de modelado conceptual capacidades de expresar la hipertexto inherente a las aplicaciones web. Consta de cinco modelos, que se especifican a continuación:

- El Modelo de Objetos define la estructura y las relaciones estáticas entre clases identificadas en el dominio del problema.

- El Modelo Dinámico describe las posibles secuencias de servicios y los aspectos relacionados con la interacción entre objetos.
- El Modelo Funcional captura la semántica asociada a los cambios de estado entre los objetos motivados por la ocurrencia de eventos o servicios.
- El Modelo de Navegación define la semántica navegacional asociada a las clases de los objetos del modelo. Es en este modelo se realiza un mapa de navegabilidad por cada uno, definiendo los nodos (posibles puntos de interacción con el usuario) y arcos (posibilidad de alcanzar otro nodo).
- El Modelo de Presentación captura los requisitos básicos de presentación de información. Está fuertemente basado en el modelo de navegación y permite definir, de una manera abstracta, la estructura lógica de presentación de los objetos navegacionales en la interfaz de usuario.

Esta metodología también cuenta con un entorno de desarrollo para aplicaciones web (OOWS Suite) (Escribano 2010). OOWS Suite es un entorno de desarrollo que se integra con la herramienta comercial OlivaNOVA, que implementa el proceso de desarrollo de OO-Method. El entorno consta de un editor gráfico denominado OOWS Visual Editor, el cual asocia a cada primitiva conceptual su representación gráfica o textual.

DDRE (*Design-Driven Requirements Elicitation*)

Esta propuesta para el tratamiento de requisitos es parte del proceso *Design-Driven* presentado por (Lowe and Eklund 2002). Consiste en realizar la captura, la definición y la validación de requisitos durante el proceso de diseño. El proceso que ofrece fue definido en base a un exhaustivo análisis de mejores prácticas en el desarrollo de aplicaciones comerciales para la web.

NDT (*Navigational Development Techniques*)

Es una propuesta metodológica que se mueve dentro del entorno de la Ingeniería Web. NDT se localiza en las primeras fases del proceso de desarrollo del software y se centra en una especificación de requisitos muy concreta que permite conseguir de manera sistemática los modelos que representan al sistema. El proceso de desarrollo de NDT cubre dos fases del proceso del desarrollo del software: la especificación de requisitos y el análisis. La especificación de requisitos comienza con una captura de los objetivos del sistema y a partir de ellos se definen los requisitos a cubrir para alcanzarlos. En NDT los requisitos se clasifican en (Escalona, Torres et al. 2003):

- Requisitos de almacenamiento de información, que definen qué información se va a manejar en el sistema y cómo se relacionan entre sí. NDT permite también definir nuevas naturalezas de datos, esto son las nuevas especificaciones de categorías de datos que se vayan a utilizar en el sistema.

- Requisitos de actores, en los que se definen los roles que podrán interactuar en el sistema y las relaciones que se pueden producir entre ellos.
- Requisitos funcionales, que permitirán definir la funcionalidad del sistema.
- Requisitos de interacción, que definen cómo se muestra la información, cómo se podrá navegar en el sistema y los criterios de recuperación que se ofrecen.
- Requisitos no funcionales, que recogerán otros requisitos del sistema.

Tras la definición y validación de los objetivos, en NDT se propone generar tres modelos: el modelo conceptual, que representa mediante un diagrama de clases la estructura estática del sistema; el modelo de navegación, que representa mediante un conjunto de diagramas de clases la forma en que se podrá navegar en el sistema; y el modelo de interfaz abstracta, que mediante un conjunto de prototipos evaluables permitirá mostrar cómo se va a interactuar con el sistema (Escalona, Torres et al. 2003).

NDT no contempla las fases de implementación o mantenimiento del proceso de desarrollo de aplicaciones Web, está centrado en la fase de definición de requerimientos. El proceso de definición de requerimientos se basa los diferentes roles de usuarios que pueden aparecen en el sistema (Rojo 2012).

En los últimos años, la metodología NDT ha evolucionado para permitir su uso en entornos prácticos, siendo hoy por hoy una de las mejores propuestas metodológicas para abordar el desarrollo de cualquier proyecto software, y en particular proyectos software orientados a la web. Esta metodología fue inicialmente soportada por la herramienta *NDT-Tool*, esta cayó en desuso tras las evoluciones de NDT y ahora la solución de herramientas es *NDT-Suite*. La misma cubre seis grupos de procesos: desarrollo, aseguramiento de la calidad, gestión, mantenimiento, pruebas y seguridad. El paquete de herramientas *NDT-Suite* está compuesto por las siguientes herramientas (Escalona, Torres et al. 2003):

NDT-Profile: es un conjunto de perfiles UML específicos para NDT. Esta herramienta permite tener y trabajar con todos los elementos de NDT y, bajo el entorno de *Enterprise Architect* permite trabajar fácilmente con los conceptos de la metodología.

NDT-Driver: permite tomando como entrada un proyecto elaborado mediante *NDT-Profile* ejecutar de manera automática las transformaciones definidas en la metodología NDT. A efectos prácticos, esta herramienta permite disminuir el tiempo empleado en la descripción y diseño de los modelos propios de las fases del desarrollo del software posteriores a la fase de captura de requisitos, ya que los modelos básicos obtenidos por esta herramienta proporciona al analista un punto de partida.

NDT-Quality: es una herramienta que automatiza el proceso de revisión de los modelos de un proyecto desarrollado con *NDT-Profile*. *NDT-Quality* chequea tanto la calidad de la propia metodología como reglas específicas de UML.

NDT-Prototypes: es una herramienta que genera de manera totalmente automática un conjunto de prototipos XHTML a partir de los modelos de navegación descritos en la fase de análisis de un proyecto desarrollado con *NDT-Profile*. A efectos prácticos, esta herramienta permite disminuir el tiempo empleado en la construcción de una aplicación web ya que proporciona al equipo de desarrollo un punto de partida a la hora de llevar a cabo la construcción del sistema.

NDT-Glossary: es una herramienta que implementa un procedimiento automático que, a partir del modelo de requisitos de un proyecto que sigue la metodología NDT y que ha sido especificado en base a la herramienta *NDT-Profile*, genera la primera instancia del glosario terminológico de dicho proyecto.

NDT-Counter: permite aplicar la técnica de los puntos de casos de uso para estimar el esfuerzo para desarrollar un proyecto con NDT.

NDT-Report: genera una serie de documentos PDF a partir de *NDT-Profile*. Es una herramienta que se encuentra integrada como un conjunto de plantillas de NDT.

NDT-Checker: es la única herramienta de *NDT-Suite* que no está basada en el paradigma MDE. Esta herramienta incluye un conjunto de plantillas, diferente por cada elemento de NDT.

WebML (*Web Modeling Language*)

Es una metodología de modelado visual de aplicaciones web, centrada especialmente en las aplicaciones de uso intensivo de datos, separando el contenido de la información en páginas, navegación y presentación, que se pueden definir y desarrollar de forma independiente. Permite la especificación de operaciones de manipulación de datos para actualizar la aplicación así como especificar sitios web complejos a nivel conceptual (Escribano 2010).

Cuenta con cuatro perspectivas:

- Modelo Estructural, para los datos de la aplicación.
- Modelo de Hipertexto, para cada hipertexto describe qué páginas lo componen y cómo navegan.
- Modelo de Presentación, para la disposición y apariencia gráfica.
- Modelo de Personalización, para definir operaciones específicas para usuarios o grupos de usuarios, ya que se almacenan como entidades en el Modelo Estructural.

Dispone de las herramientas CASE: WebRatio y TotiiSoft que dan soporte a esta metodología. WebRatio trabaja con metamodelos, modelando los objetos de negocio mediante los estándares UML. Para esos modelos se genera automáticamente la aplicación en la arquitectura JavaEE y las fuentes de datos SQL/XML.

TotiiSoft cubre las actividades del proceso de desarrollo de aplicaciones web. Está centrado en el uso de WebML. Tiene un módulo de diseño para edición de especificaciones de estructura, hipertextos y de modelos de personalización. Módulo de Manager para la administración y evolución de la aplicación en el tiempo (Rojo 2012).

Comparando las distintas metodologías descritas en la investigación, se llega a la conclusión de que WebML propone un enfoque integral de todas las fases del proceso de desarrollo del software con una profunda descripción, mientras que *WSDM*, *HFPM*, *UWE* y *DDRE* cubren el proceso de desarrollo completo de la aplicación del software. El nivel de especificación de actividades es genérico. Por otro lado, el método NDT solo cubre las fases de modelamiento y análisis de requisitos del proceso de desarrollo del software con gran nivel de detalle (Rojo and Oliveros 2012). En la Tabla 3, se resume cómo se comportan estas metodologías a través de indicadores de usabilidad, los cuáles ayudan a alcanzar niveles detallados de usabilidad.

Tabla 3: Metodologías web que evalúan los requisitos de usabilidad.

Fuente: Elaboración propia.

Metodología Web							
Indicadores	WSDM	HFPM	UWE	WebML	NDT	DDRE	OOWS
Interfaz de usuario		x	x	x	x	x	x
Navegabilidad	x	x	x		x	x	x
Adaptabilidad	x		x	x	x	x	
Herramienta Case			x	x	x		x

Luego del análisis de la incorporación de la Ingeniería de la Usabilidad desde etapas tempranas de requisitos, es importante mantener y desarrollar los requisitos de usabilidad durante el diseño, específicamente el diseño de la arquitectura; surgiendo para ello, los patrones de usabilidad. Estos aspectos por su importancia son desarrollados a continuación.

1.5 La Ingeniería de Usabilidad en la arquitectura de software. Patrones de usabilidad

La arquitectura de software se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad y servir como guía en el desarrollo del producto de software. Los diseños arquitectónicos que se crean en una organización pueden ser reutilizados para crear sistemas distintos, lo que permite reducir costos y aumentar la calidad, sobre todo si dichos diseños han resultado previamente en sistemas exitosos (Cervantes 2010).

En el diseño de una aplicación en desarrollo se encuentran dificultades características y periódicas. Para estas, existen soluciones probadas y documentadas por multitud de programadores que se les conocen como patrones y ayudan a la aplicación o al diseñador a comunicarse mejor con el usuario. Ante problemas de usabilidad surgen los patrones de usabilidad. Estos son un aspecto central en el desarrollo de software, pues se encargan de sentar las bases de un conjunto de principios validados y establecidos para la creación de una apropiada interfaz de usuario. En este sentido el esfuerzo por desarrollar un modelo de patrones de

usabilidad está justificado y permite sumar una instancia más al proceso de automatización en el desarrollo de software (Merlino 2014).

A continuación se describen los patrones de usabilidad asociados a la característica de la ISO/IEC 25010:2011, luego de un análisis de varias bibliografías consultadas como ((Folmer and Bosch 2004), (Moreno and Sánchez-Segura 2007), (Panach Navarrete 2010)).

wizard: guía al usuario en la realización de una tarea que necesita tomar varias decisiones en una misma pantalla, mostrándole cuales son los pasos que ha realizado, así como los pasos restantes para finalizar la ejecución del asistente.

progress_indication: proporciona al usuario la información sobre lo que está haciendo el sistema.

user_profile: permite al usuario que personalice la aplicación de acuerdo a sus gustos.

undo: permite al usuario volver a un estado anterior al que se encuentra.

cancel: permite al usuario cancelar una acción en la aplicación.

standard_help: proporciona ayuda al usuario sobre cómo llevar a cabo las acciones del sistema.

multi-tasking: permite al usuario realizar varias tareas a la vez en el sistema.

shortcuts: facilita al usuario acceder en la aplicación a través de distintos atajos, es decir, diversas maneras o vías que le acortan el camino para llegar a una tarea.

commands_aggregation: proporciona al usuario la facilidad de agregar comandos a la aplicación.

different_languages: proporciona al usuario la facilidad de acceder a la aplicación a través de distintos lenguajes, opción que la aplicación debe poseer.

workflow_model: proporciona al usuario a través de un mapa del sitio la manera en que puede trabajar por la aplicación, así como el lugar en que se encuentra.

alert: evita que el usuario cometa errores en el proceso de introducción de datos o en la navegación por la aplicación.

data_validation: facilita la validación de los datos de un formulario para que el usuario acceda por los mismos sin dificultad alguna.

reuse_information: proporciona al usuario la facilidad de reutilizar información dentro de la aplicación.

different_access_methods: proporciona al usuario la facilidad de acceder a través de distintos métodos o formas a la aplicación.

multiple_views: permite al usuario acceder a varias vistas en el sistema.

Luego del análisis de estos patrones de usabilidad se describirá en el epígrafe siguiente la situación real de la usabilidad en los proyectos de la UCI. Se analizará además, los cambios realizados en las evaluaciones de usabilidad para una mejor calidad en el proceso de pruebas.

1.6 Análisis de la usabilidad en los proyectos de la UCI

En Cuba, hasta la creación de CALISOFT, no existía ninguna organización que se dedicara por completo a los temas relacionados con la calidad y pruebas de software. Esto era algo que las

propias empresas productoras de software debían garantizar. Con la creación de este centro se garantiza este servicio, el cual comenzó brindándose a los proyectos de la UCI, pero hoy se proyecta para el resto de la industria cubana de software (Capote García 2011).

A pesar de que en el DEPSW de CALISOFT se realizan evaluaciones de usabilidad aplicando el método de evaluación heurística sin intervención del usuario final, no se logra establecer en la misma, un resultado con la mayor calidad posible. Muchos de los proyectos sometidos a este servicio, no tienen concebido durante su etapa de desarrollo del producto, indicadores que los ayude a gestionar los requisitos y patrones de usabilidad como se demuestra en la entrevista a profundidad aplicada al inicio de la investigación. A mediados del 2014 se empezó a realizar las evaluaciones de usabilidad en CALISOFT por 3 ingenieros de pruebas del grupo de usabilidad, expertos en el tema, ya que antes solo la ejecutaba el especialista al frente del proceso de la evaluación en un plazo promedio de 10 días. Esto ha traído como consecuencia que se encuentre más cantidad de NC de usabilidad con mayor precisión en el menor tiempo posible, siendo este de 4 días promedio.

De las diez evaluaciones de usabilidad tomadas como muestra en los años 2012 y 2013 se puede decir que se detectó un total de 245 NC en la primera iteración, clasificadas las mismas como: error de interfaz, error de idioma y formato (Ver Figura 7).

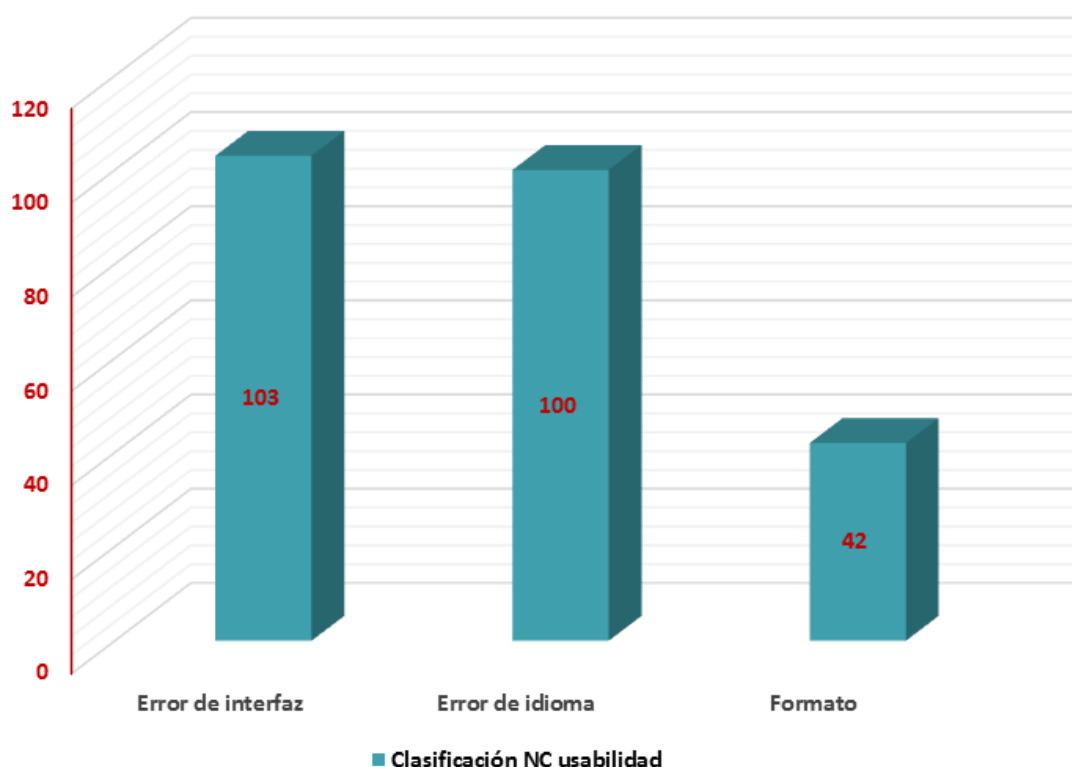


Figura 7: Clasificación de las NC de usabilidad detectadas en el período 2012-2013.

Fuente: Elaboración propia.

En el 2014 se refinó la clasificación de las NC en: error de interfaz, error de idioma, formato, navegabilidad, compatibilidad, legibilidad, visibilidad, calidad del contenido, control de usuario y recomendación. De los ocho proyectos tomados como muestra para la ejecución de las evaluaciones de usabilidad en la primera iteración, se detectó 168 NC (Ver Figura 8).

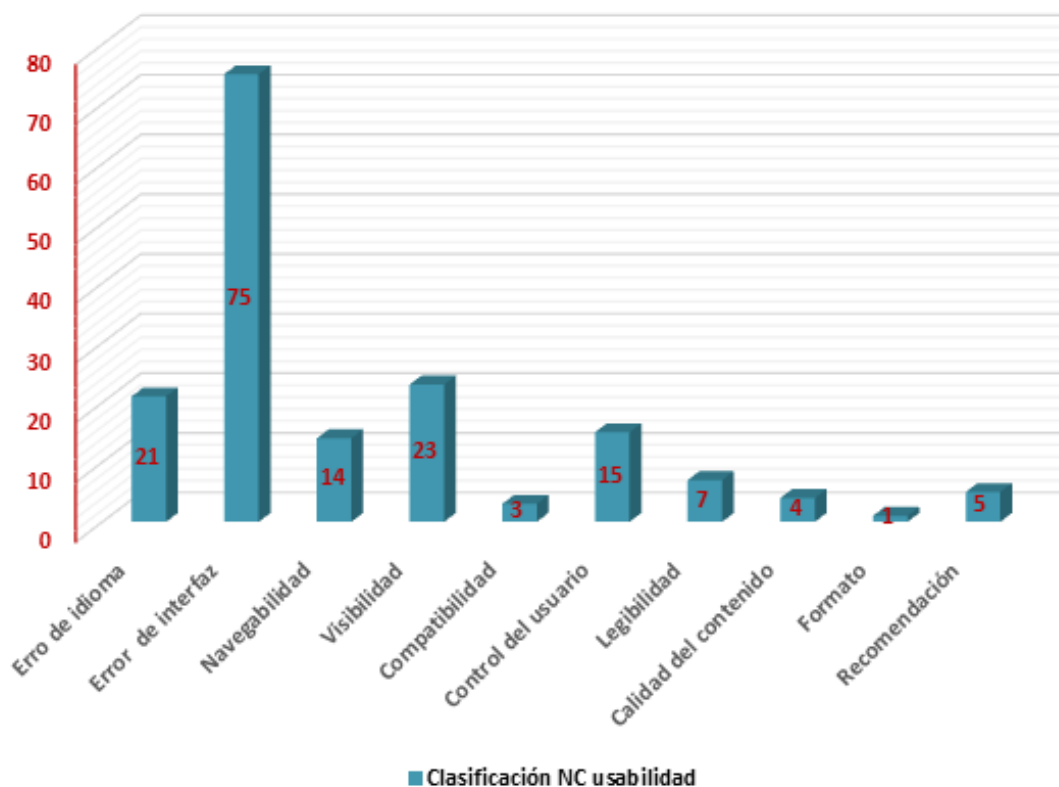


Figura 8: Clasificación de las NC de usabilidad detectadas en el 2014.

Fuente: Elaboración propia.

Estas clasificaciones, refinadas de un período a otro, han dado una visibilidad de los principales problemas de usabilidad que existen hoy en día. De acuerdo a las mismas se enfatiza en la necesidad de incorporar técnicas y herramientas en la gestión de los requisitos de usabilidad, así como patrones de usabilidad desde temprana etapa de desarrollo del producto de software para erradicar la mayor cantidad de NC de usabilidad.

Conclusiones

Después de un análisis del estado del arte, destacándose la característica usabilidad de la ISO/IEC 25010:2011, se puede concluir que es la más completa dado al conjunto de elementos complementarios, terminologías y actividades. La usabilidad por concepto mide elementos objetivos y subjetivos, en la cuál es difícil su gestión a nivel de requisitos y diseño. A pesar de eso, hay varios autores que proponen su inclusión en el proceso de desarrollo del software, donde se constata la importancia de considerar técnicas y herramientas en los requisitos de usabilidad

desde etapas tempranas en su desarrollo. Además, es esencial tener en cuenta los patrones en el diseño, por las mejoras que estos brindan en la arquitectura de software.

La interfaz de usuario, navegabilidad y adaptabilidad son elementos que se utilizan para evaluar la usabilidad en la web. De acuerdo a las bibliografías consultadas se concluye la relevancia de contar con la metodología NDT y su herramienta NDT-Suite como apoyo a la solución a proponer. Las evaluaciones de usabilidad realizadas en la UCI, han dado una claridad de dónde radican las mayores deficiencias de usabilidad a través de la clasificación de las NC, de ahí la importancia de corregir estos problemas incorporando al proceso de desarrollo del software la Ingeniería de Usabilidad.

CAPÍTULO II: PROCEDIMIENTO PARA INCORPORAR LA INGENIERÍA DE USABILIDAD DESDE ETAPAS TEMPRANAS

Introducción

En el presente capítulo se realiza una descripción de la propuesta de solución donde se sintetiza en la clasificación de roles y responsabilidades. Se propone el rol del diseñador y del arquitecto de información dentro de esta clasificación, por la importancia que estos ameritan en las etapas por la que transcurre la IU. Se determinan los artefactos de entrada y salida, actividades, técnicas, metodologías y herramientas para apoyar la flexibilidad de la gestión de los requisitos y patrones de usabilidad durante proceso de desarrollo de un producto software. Además, se relacionan los atributos, propiedades y patrones de usabilidad que aportan en la solución propuesta.

2.1 Diagnóstico inicial

Con el objetivo principal de evaluar el estado en que se encuentran la gestión de los requisitos de usabilidad en los proyectos productivos se realizó un diagnóstico inicial a partir de una encuesta (Ver Anexo 2). En el estudio llevado a cabo participaron 17 personas con diferentes roles de distintos proyectos como: desarrolladores, analistas de software, jefe de proyecto, diseñadores; con experiencias en desarrollo de proyectos, ingeniería de requisitos y en temas de usabilidad. Los mismos con procedencia de las siguientes instituciones del país: centros de desarrollo de la UCI, XETID, DESOFT, SOFTEL, ETECSA, para lograr identificar cuántos de ellos gestionan requisitos de usabilidad en sus proyectos; qué roles participan en el desarrollo de esta actividad; de acuerdo al rol que desempeña en el proyecto, qué responsabilidades considera que no deben faltar para realizar dicha tarea; y qué técnica(s) utilizan para el desarrollo de los requisitos de usabilidad.

De la encuesta realizada a una muestra aleatoria de 17 especialistas, el 35.29% planteó que sí gestionaban los requisitos de usabilidad en sus proyectos, y el 64.71% afirmó lo contrario. A pesar de que el 35.29% manifestó realizar requisitos de usabilidad, se pudo comprobar que no los efectuaban correctamente, debido al limitado conocimiento de la característica usabilidad de la calidad de software; solo el 33.3% tenía en cuenta técnicas para el desarrollo de los requisitos. En cuanto a los roles que deben participar en la definición de los requisitos de usabilidad, el 88.24% de los encuestados refirió la necesidad de introducir el rol de diseñador en los proyectos, para erradicar en gran medida los problemas del diseño de la interfaz de los software. El 92.2% comunicó la importancia de incorporar el rol de arquitecto de información a parte del de diseñador por las responsabilidades que este tiene en la arquitectura de usabilidad. Por otra parte, se afirmó que el 95% no utiliza patrones de usabilidad para mejorar la misma en el desarrollo de los productos de software.

Esta encuesta tuvo como resultado la necesidad e importancia de introducir la gestión de los requisitos y patrones de usabilidad en el proceso de desarrollo del software, para que los mismos terminen con la calidad requerida sin tener que ser rediseñados, o hacer cambios en ellos cuando ya estos estén culminados. Se considera importante el tema de investigación que se propone por sus aportes en los proyectos productivos.

2.2 Propuesta del procedimiento

La constante evolución de las tecnologías para realizar evaluaciones de usabilidad hace que cada día aumenten la necesidad de gestionar requisitos de usabilidad en los proyectos productivos. Proveer a los analistas de software de metodologías, técnicas y herramientas que permitan acelerar el desarrollo de esta actividad constituye un salto cualitativo en el desarrollo de un producto de software exitoso.

Para la concepción de la solución se tiene en cuenta como principios y características los siguientes:

- Integrar las distintas etapas de la Ingeniería de Usabilidad en el proceso de desarrollo del software de manera flexible.
- Facilitar el trabajo de los analistas de software permitiendo mejoras en los productos de software de la sociedad.
- Lograr de una manera integrada cerrar el proceso de desarrollo de los requisitos de usabilidad en un producto.
- Disminuir el esfuerzo en la obtención de conocimientos a partir de la incorporación de la Ingeniería de Usabilidad a nivel de requisitos y de patrones de usabilidad.
- Brindar información para que las autoridades competentes tomen estrategias y medidas a aplicar en función de elevar la calidad del producto de software.

En la Figura 9 se presenta la relación que existe entre el proceso de desarrollo del software con las etapas de la Ingeniería de Usabilidad.

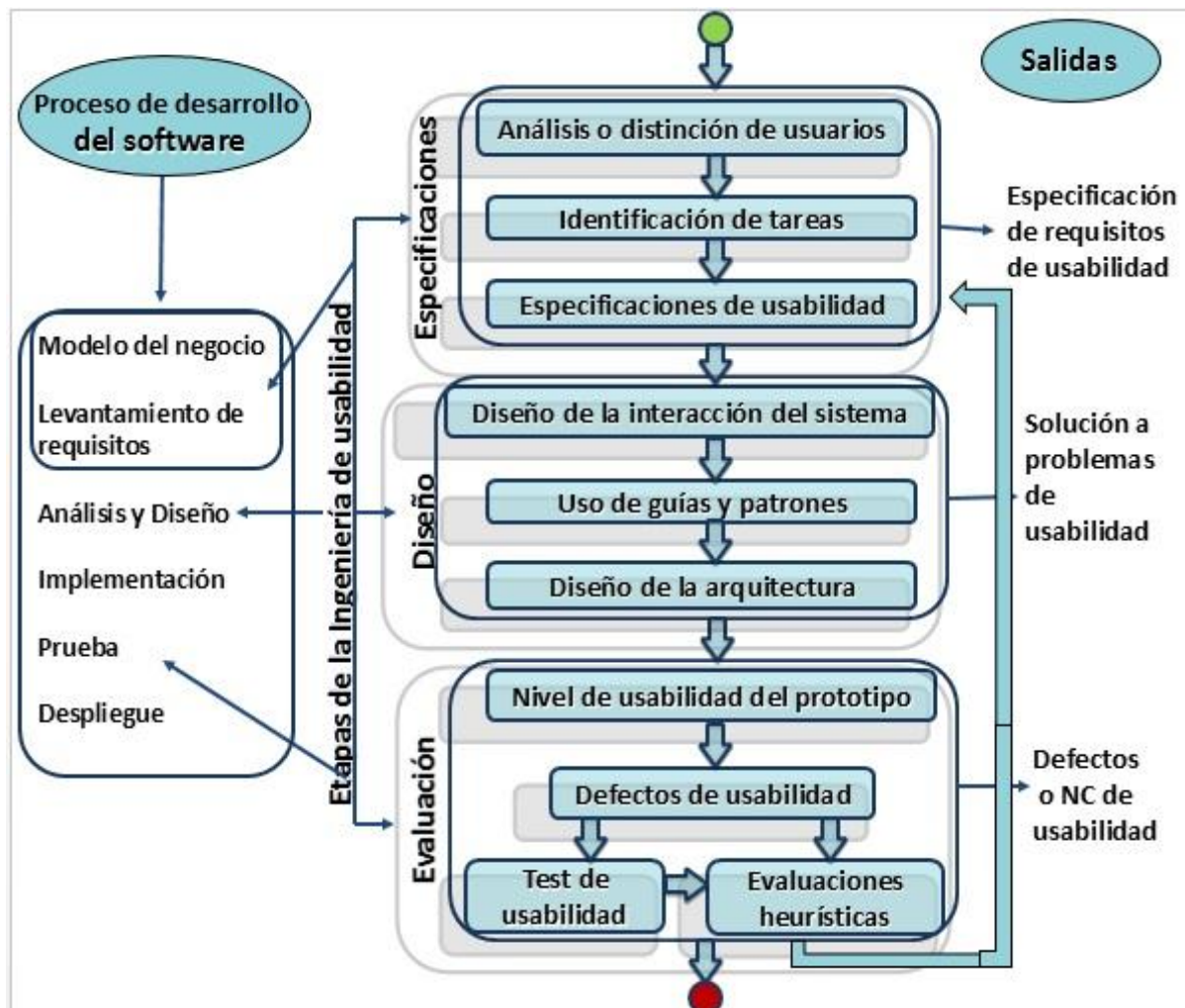


Figura 9: Etapas de la Ingeniería de Usabilidad insertada en el proceso de desarrollo del software.

Fuente: Elaboración propia.

A través de una imagen es posible representar una idea, pero en ocasiones es necesario transmitir detalles mediante un texto que no es posible a través de un gráfico. En ese caso se encuentran las fases de especificaciones, diseño y evaluación de la IU que se pueden insertar en la de levantamiento de requisitos, análisis y diseño, y pruebas, del proceso de desarrollo del software respectivamente. En cada fase de la Ingeniería de Usabilidad se encuentran las tareas más importantes, teniendo como artefactos de salidas: la especificación de requisitos de usabilidad, solución a problemas de usabilidad y defectos o NC de usabilidad. Todas estas salidas garantizan una mayor calidad y usabilidad del producto de software permitiendo el éxito del mismo. Luego se describirá gráfica y textual las fases de especificaciones y diseño del procedimiento propuesto en los epígrafes 2.3 y 2.4.

2.3 Descripción gráfica y textual de la incorporación de la Ingeniería de Usabilidad a nivel de requisitos

Un aspecto fundamental de este trabajo es proponer el desarrollo de los requisitos de usabilidad. Las técnicas y artefactos entregables ayudan a verificar que las actividades se cumplan

continuamente en el proceso de desarrollo del software (Ver Figura 10). Es importante aclarar que cada actividad genera una serie de artefactos que son insumo para la actividad siguiente. El proceso se concibe de manera cíclica e iterativa, en el cual se van depurando gradualmente la característica usabilidad de la aplicación.

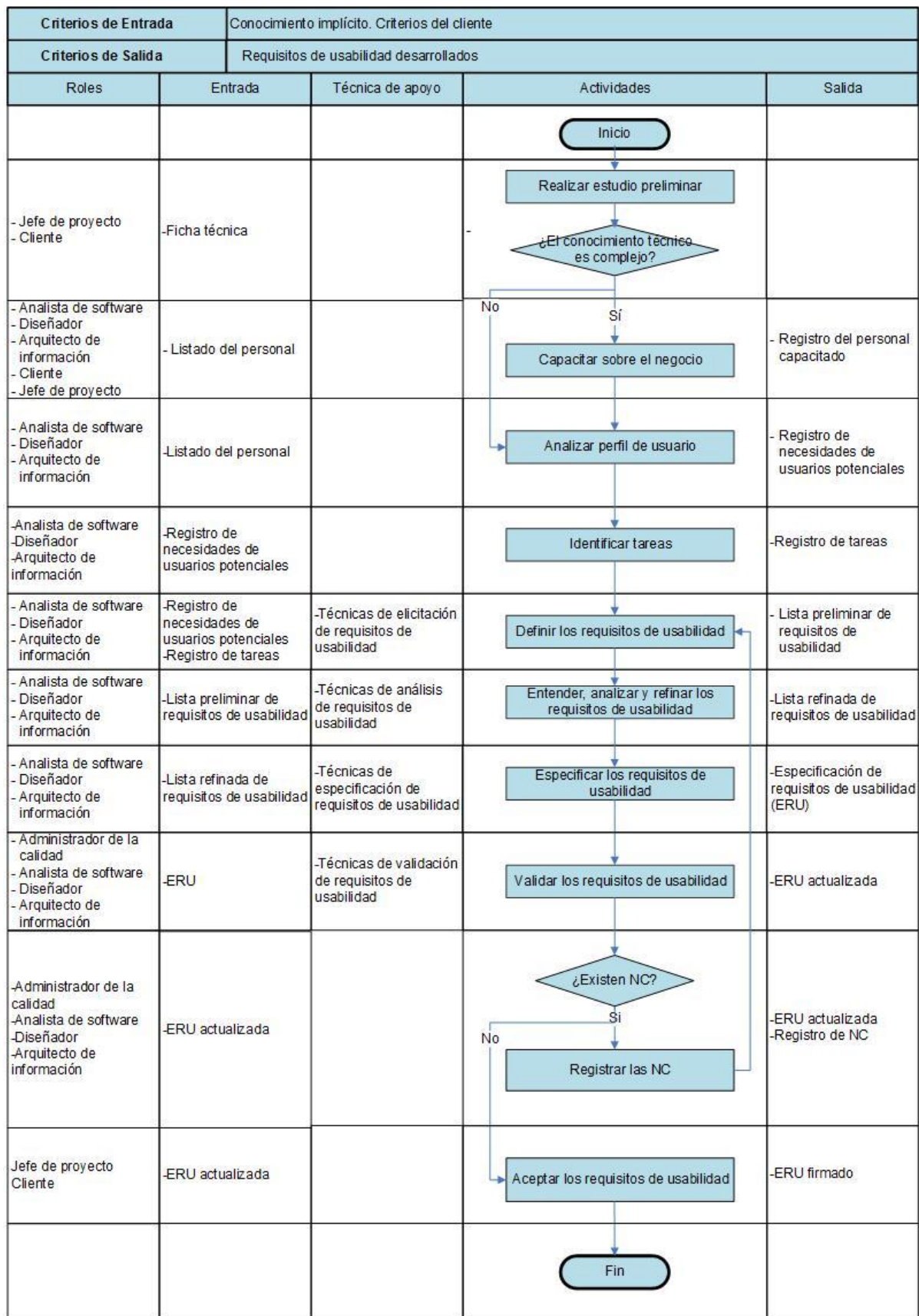


Figura 10: Descripción gráfica del desarrollo de los requisitos de usabilidad.

Fuente: Elaboración propia.

A continuación se describen las actividades con sus artefactos de entrada o salida en el desarrollo de los requisitos de usabilidad:

Realizar estudio preliminar: el cliente entrega una ficha técnica al Jefe de proyecto que sirve como entrada para conocer la visión general del proyecto, análisis de factibilidad, proyecto técnico.

Pregunta: ¿necesita conocimiento técnico muy complejo?, en caso de que la respuesta sea sí, se capacita sobre el negocio, en caso contrario realiza se analiza el perfil de usuario.

Capacitar sobre el negocio: el Jefe de proyecto entrega el listado del personal a capacitar sobre el negocio. En esta actividad el cliente imparte capacitación sobre el negocio a informatizar durante todas las etapas del desarrollo, realizando la función de especialista funcional; se tiene como artefacto de salida un listado del personal capacitado.

Analizar perfil de usuario: puede usarse para esta actividad un modelo de usuarios, donde el analista de software, diseñador y arquitecto de información definen el perfil de usuarios potenciales a partir del listado del personal entregado. Se describen los factores más relevantes de impacto que inciden sobre la usabilidad del producto, saliendo el registro de necesidades de usuarios potenciales y clasificando los mismos en principiantes, medios y avanzados.

Identificar tareas: de acuerdo al registro de necesidades de usuarios potenciales, entrada en esta actividad, se describen las tareas que realizan los usuarios, los flujos de trabajo y se levantan las necesidades de información que requieren para realizar sus trabajos. Además, se detalla la personalización y se declaran atajos para facilitar las tareas de operaciones específicas que pueden ser complejas, al usuario de manera rápida. Se obtiene como artefacto de salida el registro de tareas.

Definir los requisitos de usabilidad: a través del registro de necesidades de usuarios potenciales y del registro de tarea; el analista de software, diseñador y arquitecto de información especifican los objetivos cualitativos y cuantitativos de la usabilidad. Para desarrollar esta actividad se apoyan en las técnicas adecuadas para la elicitación de los requisitos de usabilidad, teniendo en cuenta en la definición de estos elementos como: organización y representación de los contenidos y servicios de la página o pantalla a través de definición de la taxonomía, el sistema de etiquetado y de navegación de la aplicación. Además, indicadores que pudieran establecerse en la pauta de diseño: cromática (color), iconográfica, tipográfica, apoyo gráfico, resoluciones de pantalla, grillas (banner, menú, imágenes y pie de página o pantalla), el logotipo e isotipo de la aplicación; los elementos visuales atractivos que transmitan una idea específica; la forma visual de la interfaz de usuario, esquema de pantallas, modelos de interfaz gráfica. También se definen en los requisitos de usabilidad la validación de datos, funciones complejas con recuperación de información, mensajes de alertas, personalización de las funciones según las habilidades del usuario, entre otras. Se obtiene como resultado de esta actividad, una lista preliminar de requisitos de usabilidad.

Entender, analizar y refinar los requisitos de usabilidad: conforme con la lista preliminar de requisitos de usabilidad, el arquitecto de información, el diseñador y el analista de software realizan un análisis de los requisitos de usabilidad de proyectos anteriores que puedan reutilizarse, apoyándose en las técnicas adecuadas para el análisis de los requisitos de usabilidad. Se tiene como salida de esta actividad la lista refinada de requisitos de usabilidad.

Especificar los requisitos de usabilidad: partiendo de la lista refinada de requisitos de usabilidad, se revisa y especifica los requisitos de usabilidad (ERU) por el arquitecto de información, el diseñador y el analista de software, garantizando que se cumpla adecuadamente la descripción de los requisitos con las subcaracterísticas de usabilidad, apoyándose en las técnicas adecuadas para especificar los mismos. Se tiene de esta actividad como artefacto de salida la ERU.

Validar los requisitos de usabilidad: se valida la especificación de requisitos de usabilidad como resultado de las técnicas aplicadas para validar las mismas en conjunto con el administrador de la calidad, el arquitecto de información, el diseñador y el analista de software, concluyendo con la ERU actualizada.

Pregunta: ¿existen NC?, en caso positivo se registran las mismas en el registro de NC, se actualiza la ERU y se pasa a definir nuevamente los requisitos de usabilidad, en caso contrario, se realiza la aceptación de los requisitos de usabilidad.

Aceptar los requisitos de usabilidad: se aceptan los requisitos de usabilidad por ambas partes: el cliente y el Jefe de proyecto, saliendo como artefacto luego de esta actividad, la ERU firmada.

Estos requisitos de usabilidad pueden ser trabajados y desarrollados a través de la metodología de desarrollo web NDT y su herramienta case, para enriquecer el proceso de desarrollo del software a partir de la introducción temprana de requisitos web en un producto. Haciendo una mejora de los trabajos anteriores que incurren en este tema y mencionados en el capítulo I, se propone la misma por las siguientes razones:

- Ofrece una interfaz sencilla para la comunicación con el usuario final.
- Se basa en un conjunto de metamodelos, que resultan transparentes para el grupo de desarrollo, que da soporte al proceso de desarrollo.
- Cuida la trazabilidad de los requisitos desde la captura hasta el análisis, ofreciendo un proceso sistemático de desarrollo basado en transformaciones formales descritas con QVT (*Query/View/Transformation*) que lleva hasta la implementación.
- Ha tenido y está teniendo una gran aplicabilidad práctica en empresas y proyectos reales.
- El proceso de definición, captura de requisitos y objetivos que propone se basa principalmente en plantillas o patrones, que no son más que tablas con una estructura predefinida que describen a los requisitos.

Para incorporar requisitos de usabilidad en el proceso de desarrollo del software es preciso contar con un grupo de técnicas y herramientas que apoyen el proceso de desarrollo de requisitos. Estas

han sido utilizadas de forma clásica y dado a sus particularidades se precisará a continuación los criterios de selección definidos para optar por las mismas.

2.3.1 Técnicas y herramienta propuestas para el desarrollo de requisitos de usabilidad

No existen técnicas estandarizadas y estructuradas que ofrezcan un marco de desarrollo para llevar a cabo los procesos de la ingeniería de requisitos de usabilidad durante la construcción de un sistema. La selección de las mismas y el éxito de los resultados que se obtienen dependen en gran medida, tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participan.

En aras de disminuir la complejidad del desarrollo de los requisitos de usabilidad se recomienda el uso de algunas técnicas descritas en el capítulo I como: entrevistas, tormenta de ideas (brainstorming); mapas conceptuales; prototipos; extracción de fuentes como documentos del negocio, estándares o pautas establecidas en la organización cliente; encuestas; así como la herramienta lista de chequeo (Ver Tabla 4). Las técnicas propuestas están consideradas en las distintas fases que contiene la Ingeniería de Requisitos, y su selección está dada por los siguientes criterios:

- Visión del problema.
- Facilidad de aplicación.
- Facilidad de aprendizaje.
- Bajo costo de aplicación.
- Ventajas entre otra de las técnicas descrita.
- Popularidad.
- Precisión.
- Ayuda inestimable.

Tabla 4: Técnicas propuestas para el desarrollo de los requisitos de usabilidad.

Fuente: Elaboración propia.

Técnicas	Elicitación	Análisis	Especificación	Validación
Entrevistas	x			
Tormenta de ideas (Brainstorming)	x	x		
Mapas conceptuales			x	
Encuesta	x			
Extracción de fuentes como documentos del negocio, estándares o pautas establecidas en la organización cliente	x	x		
Prototipos		x	x	
Lista de chequeo				x

Lista de chequeo

Para validar la verificabilidad de los requisitos se elaboró una lista de chequeo. La misma posee varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo presente su pertinencia, además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar. El objetivo general de la lista de chequeo es validar la usabilidad de los productos.

Esta herramienta manual ha sido confeccionada para guiar a analistas, desarrolladores, administradores de la calidad o expertos técnicos en la verificación y validación de la usabilidad. Permite recoger los puntos eficientes e ineficientes que tienen los indicadores a evaluar, aplicados a cada una de las revisiones de usabilidad que se desarrollen. La misma posee un carácter flexible para el administrador de calidad o el desarrollador, teniendo en cuenta que pueden surgir modificaciones e inclusiones. La lista está enfocada a tres áreas fundamentales (Ver Tabla 5) (Hernández and Machado 2010):

Estructura del documento: abarca todos los aspectos definidos por el expediente de proyecto o el formato establecido por el proyecto.

Elementos definidos por la metodología: abarca todos los indicadores a evaluar según la metodología y el artefacto a revisar.

Semántica del documento: contempla todos los indicadores a evaluar respecto a la ortografía y redacción.

Para cada área se elaboró un grupo de preguntas, que permite verificar la correctitud y completitud del artefacto a revisar, en este caso, la especificación de requisitos en cualquiera de sus variantes. Los elementos que se tuvieron en cuenta fueron:

Peso: define si el indicador a evaluar es crítico o no.

Evaluación (Eval): es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”).

NP (No Procede): se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Cantidad de elementos afectados: especifica la cantidad de errores encontrados sobre el mismo indicador.

Comentario: especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

Las preguntas relacionadas con la validación de los requisitos de usabilidad aparecerán como indicadores en el área Elementos definidos por la metodología y serán elaboradas a partir de los criterios vistos anteriormente.

Tabla 5: Estructura de la lista de chequeo.

Fuente: (Hernández and Machado 2010).

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Elementos definidos por la metodología					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Semántica del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios

Es una buena práctica elaborar las preguntas en base a las consideraciones antes mencionadas, pues el objetivo de esta investigación no es dar una lista de chequeo estática que le permita comprobar que los requisitos especificados son válidos. Se propone incorporar a la lista de chequeo de requisitos (Ver Tabla 6) elaborada por el DEPSW de CALISOFT estos indicadores de usabilidad definidos de acuerdo a las subcaracterísticas de calidad basada en la ISO/IEC 25010: 2011 propuesta en esta investigación.

Tabla 6: Indicadores incorporados a la lista de chequeo de requisitos del DEPSW de CALISOFT.

Fuente: Elaboración propia.

Elementos definidos por la metodología					
Peso	Indicadores a evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Se han descrito los requisitos de usabilidad teniendo en cuenta la subcaracterística capacidad para reconocer su adecuación (cabeceras, longitud de página o pantalla/contenido, enlaces y navegabilidad)?				
crítico	2. ¿Se han descrito los requisitos de usabilidad teniendo en cuenta la subcaracterística capacidad de aprendizaje (título/memorización, mensajes y ayuda)?				
crítico	3. ¿Se han descrito los requisitos de usabilidad teniendo en cuenta la subcaracterística capacidad para ser usado (búsqueda, instalación y configuración/recursos/actualización y compatibilidad)?				
	4. ¿Se han descrito los requisitos de usabilidad teniendo en cuenta la subcaracterística protección contra errores de usuario (errores y etiquetado)?				
crítico	5. ¿Se han descrito los requisitos de usabilidad teniendo en cuenta la subcaracterística estética de la interfaz de usuario (esquemas de página, tipografía del texto/color/diseño gráfico)?				
crítico	6. ¿Se han descrito los requisitos de usabilidad teniendo en cuenta la subcaracterística accesibilidad?				

Los indicadores dependen de los aspectos tenidos en cuenta en la especificación de requisitos de usabilidad de los productos de software.

Los indicadores de la lista de chequeo evalúan la especificación de requisitos de usabilidad descritos por el proyecto, identificando el nivel de detalles de usabilidad que contiene el producto desarrollado. De ahí la importancia de describir detallada y coherentemente cada uno de los requisitos para evitar retrasos y problemas de usabilidad en el producto final. Luego se realiza la propuesta de los requisitos de usabilidad a considerar en las organizaciones que desarrollan aplicaciones informáticas.

2.3.2 Propuesta de requisitos de usabilidad

Esta investigación propone los requisitos de usabilidad que como mínimo un software debe cumplir, para garantizar un nivel adecuado de comprensibilidad de sus servicios por parte de los usuarios. El objetivo es definir y especificar por el analista desde etapas iniciales del proyecto, el desarrollo de requisitos de usabilidad de manera general, para que estos puedan ser utilizados identificando los problemas principales en los productos y garantizar así, una correcta evaluación de usabilidad. Estos requisitos de usabilidad pueden ser enriquecidos por la experiencia y acuerdo de los especialistas en el tema.

La propuesta de requisitos de usabilidad está descrita según las subcaracterísticas de la ISO/IEC 25010: 2011. Los requisitos están agrupados por elementos de usabilidad como son: cabeceras, longitud de página o pantalla /contenido, enlaces, navegabilidad/formularios o diálogos, título/memorización, mensajes, ayuda, instalación y configuración/recursos/actualización, búsqueda, compatibilidad, errores, etiquetado, esquemas de página o pantalla, tipografía del texto/color/diseño gráfico para una mejor localización y comprensión de los mismos. A continuación se describen los requisitos de usabilidad propuestos:

Capacidad para reconocer su adecuación

- cabeceras

1. Indicar al usuario en qué parte de la estructura de la aplicación se encuentra, es decir si muestra 'migas de pan'.
2. Utilizar encabezados relacionados adecuadamente con la información que se describe en cada página o pantalla de la aplicación.
3. Utilizar el *favicon* (ícono favorito) en la barra de dirección de la aplicación.

- longitud de página o pantalla /contenido

4. Utilizar preferiblemente el uso de páginas o pantallas cortas para no abrumar al usuario.
5. Presentar o señalar la fuente utilizada en la información o contenido.
6. Permitir usar la aplicación sin necesidad de desplazar la barra horizontal.

7. Elaborar en el mismo idioma todo el contenido, en caso de utilizar otro idioma el formato de este debe ser distinguible.
 8. Utilizar un lenguaje en el contenido más asequible al usuario que al técnico.
 9. Utilizar gráficos para facilitar el entendimiento del contenido.
- enlaces
 10. Incluir enlaces referenciados para ampliar la información o contenido.
 11. Diferenciar los enlaces internos y externos.
 12. Usar preferentemente enlaces de textos, en vez de gráficos.
 13. Diferenciar los vínculos ya visitados de los no visitados.
 14. Identificar los enlaces fácilmente sin necesidad de utilizar el cursor para verificar si son vínculos (Ejemplo: están subrayados).
 - navegabilidad/formularios o diálogos
 15. Ubicar en las páginas o pantallas internas de la aplicación un acceso a la página o pantalla de inicio en una zona visible y reconocible.
 16. Ubicar en cada página o pantalla puntos de salida que permitan al usuario abandonar la tarea actual en la que se encuentre.
 17. Ubicar el menú de navegación en un lugar destacado y sin mucha profundidad.
 18. Ofrecer mapa del sitio, incluyendo en las páginas internas un acceso al mismo.
 19. Utilizar diálogos o formularios similares para tareas equivalentes.

Capacidad de aprendizaje

- título/memorización
 20. Incluir en cada página o pantalla de la aplicación un título, que facilite su identificación por los motores de búsqueda.
 21. Nombrar fácilmente la dirección de la aplicación para que sea recordada por el usuario.
- mensajes
 22. Disponer de mensajes precisos, claros e insistentes durante una operación que oriente al usuario en cada procedimiento.
- ayuda
 22. Brindar a los usuarios una ayuda en la aplicación.
 23. Brindar información de contacto relacionada con el equipo de soporte.

Capacidad para ser usado

- búsqueda
 24. Incluir el buscador en todas las páginas o pantallas internas de la aplicación.
 25. Cubrir las búsquedas en todo el sitio, no en una porción de él.
 26. Garantizar etiquetas de metadatos básicas como: palabras claves, título de la publicación, autor, descripción.

- instalación y configuración/recursos/actualización

27. Informar al usuario de los programas de software adicionales requeridos.
28. Garantizar listas de preguntas frecuentes, las cuales deben ser actualizadas periódicamente.
29. Permitir al usuario la autenticación de manera sencilla.
30. Habilitar el botón de atrás del navegador.
31. Facilitar aceleradores o atajos en la aplicación.
32. Facilitar el acceso a la información mediante la descarga e impresión.
33. Permitir la impresión de la página o contenido sin perder información.

- compatibilidad

34. Garantizar la compatibilidad de la aplicación en múltiples navegadores.

Protección contra errores de usuario

- errores

35. Mostrar una página de error personalizada si se busca una página que no existe.
Informar en la página de error personalizada cuáles son las acciones correctoras.
36. Permitir volver a la situación anterior tras una acción relevante.
37. Permitir la función de la página cuando se desconecta o no se soportan scripts.
38. Permitir la búsqueda al entrar errores tipográficos y ortográficos en el buscador.
39. Ofrecer indicaciones para completar campos problemáticos.
40. Ofrecer la posibilidad de seleccionar la información de una lista en situaciones donde pueda producir errores de escritura.

- etiquetado

41. Nombrar etiquetas de los menús, botones y las cajas de diálogo adecuadamente aplicado a la acción, no general (Ej.: utilizar “Enviar” en vez de “OK”...)

Estética de la interfaz de usuario

- esquemas de página o pantalla

42. Ubicar en la parte superior de manera clara la información más importante.
43. Ubicar el logo de la organización con total legibilidad en el mismo lugar de todas las páginas o pantallas de la aplicación.

- tipografía del texto/color/diseño gráfico

44. Mantener una tipografía y estética coherente en toda la aplicación.
 - Utilizar un tamaño de letra legible.
 - Permitir incrementar y disminuir el tamaño de la letra sin que se quiebre el esquema de la aplicación.
 - Combinar adecuadamente los colores de la aplicación de acuerdo a las características de los usuarios.

45. Mantener la misma tipografía en la aplicación para todos los navegadores.
46. Garantizar contraste de color entre fondo, texto e ilustraciones.
47. Utilizar una interfaz adecuada con características similares a las de su tipo de aplicación.
48. Garantizar una interfaz adaptable a cualquier dispositivo.

Accesibilidad

51. Permitir que el usuario pueda interactuar con la aplicación independientemente del dispositivo de entrada o salida utilizado.
52. Proporcionar formatos alternativos para el contenido visual y enlaces de texto para cada región activa de un mapa de imagen.
53. Brindar una interfaz multilingüe atendiendo al idioma de los usuarios.
54. Permitir al usuario controlar cambios en el contenido (así pueden pausar o detener el movimiento, parpadear, desplazarse o auto actualizar los objetos, pantallas o páginas).
55. Permitir acceder al contenido cuando las imágenes estén deshabilitadas o no tengan soporte.
56. Proporcionar enlaces descriptivos para usuarios no videntes.
57. Proporcionar un texto equivalente para todo elemento no textual, tales como imágenes.
58. Proporcionar presentaciones multimedia sincronizadas con sus subtítulos.
59. Poseer un documento estructurado para que pueda ser leído con o sin una hoja de estilo, utilizando adecuadamente los *tags* de HTML.

Estos requisitos mejoran en gran medida la usabilidad de una aplicación de software. Después de haberse realizado esta propuesta se hará hincapié en el procedimiento para incorporar la Ingeniería de Usabilidad al proceso de desarrollo del software a través de patrones de usabilidad que ayuden a la arquitectura de software.

2.4 Descripción gráfica y textual de la incorporación de la Ingeniería de Usabilidad a nivel de arquitectura de software

En la Figura 11 se describe la incorporación de la Ingeniería de Usabilidad a nivel de la arquitectura de software. Aquí se observa la transformación de muchos de los artefactos de entradas en salidas, dado a la información necesaria para llevar a cabo las actividades definidas en la solución propuesta.

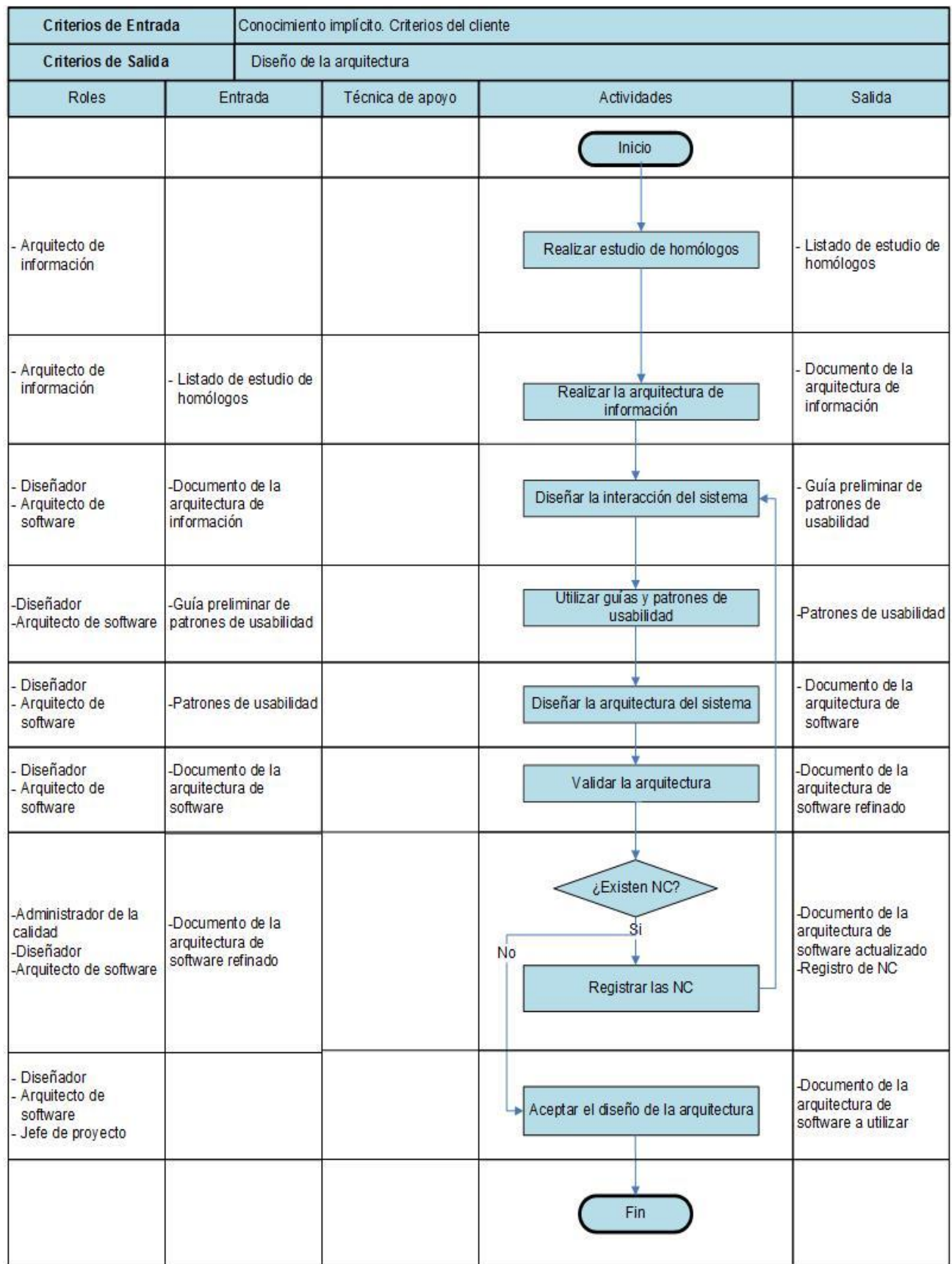


Figura 11: Descripción gráfica de la arquitectura de software.

Fuente: Elaboración propia.

A continuación se describen las actividades con sus artefactos de entrada o salida en la arquitectura de software.

Realizar estudio de homólogos: el arquitecto de información realiza un estudio de homólogos partiendo de la situación real que se desea. Aquí se hace un análisis de las aplicaciones existentes con objetivo similar, observando las ventajas y las desventajas de estas, para no aplicarlas a la que se desea desarrollar. Se tiene como resultado de esta actividad un listado de estudio de homólogos.

Realizar la arquitectura de información: el arquitecto de información partiendo del listado de estudio de homólogos realiza la arquitectura de información, es decir, distribuye o diseña donde se mostrará la organización y representación de los contenidos y servicios a través de definición de la taxonomía, el sistema de etiquetado y de navegación de la aplicación; se tiene como artefacto de salida el documento de la arquitectura de información.

Diseñar la interacción del sistema: el diseñador y el arquitecto de software conforme con el documento de la arquitectura de información definen la interacción del sistema a partir del documento de la arquitectura de información, saliendo una guía preliminar de los patrones de usabilidad a utilizar.

Utilizar guías y patrones de usabilidad: de acuerdo a la guía preliminar de los patrones de usabilidad, entrada en esta actividad, el arquitecto de software y el diseñador utilizan las guías y patrones de usabilidad, obteniéndose como artefacto de salida los patrones de usabilidad que se requiere en el desarrollo del producto.

Diseñar la arquitectura del sistema: apoyándose en los patrones de usabilidad, el arquitecto de software y el diseñador realizan el diseño de la arquitectura del sistema. Se tiene como salida de esta actividad el documento de la arquitectura de software.

Validar la arquitectura: el arquitecto de software, el diseñador en conjunto con el administrador de la calidad validan la arquitectura a través del documento de la arquitectura de software que fueron salida de la actividad anterior, concluyendo con el documento de la arquitectura de software refinado.

Pregunta: ¿existen NC?, en caso positivo se registran las mismas en el registro de NC, se actualiza el documento de la arquitectura de software y se pasa a diseñar nuevamente la interacción del sistema, en caso contrario, se realiza la aceptación del diseño de la arquitectura.

Aceptar el diseño de la arquitectura: el Jefe de proyecto en conjunto con el diseñador y el arquitecto de software acepta el diseño de la arquitectura saliendo luego de esta actividad el documento de la arquitectura de software a utilizar.

La incorporación de los patrones de usabilidad en la arquitectura de software guardan cierta relación con los atributos de usabilidad de la NC/ISO 25010: 2011. A continuación se muestra la interacción que estos poseen, para el éxito de una aplicación de software.

2.4.1 Incorporación de la Ingeniería de Usabilidad en la arquitectura de software. Patrones de usabilidad y su relación con los atributos y propiedades de usabilidad

Los patrones de usabilidad son un conjunto de buenas prácticas de software, las cuales hacen que una aplicación tenga una interacción acorde a las características y expectativas del usuario (Ferré 2003).

Cabe mencionar que los patrones de usabilidad no proporcionan una solución software concreta para ser incorporada a la arquitectura del software, simplemente sugieren algún mecanismo abstracto que podría utilizarse para mejorar la usabilidad. En esta investigación se estará trabajando con los siguientes patrones de usabilidad: *undo*, *alert*, *standard_help*, *cancel*, *multi_tasking*, *wizard*, *progress_indication*, *user_profile*, *data_validation*, *different_languages*, *shortcuts (key and task)*, *different_access_methods*, *multiple_views*, *workflow_model* descritos en el capítulo anterior.

La relación entre los atributos, las propiedades y los patrones de usabilidad se describen en detalle en la Figura 12, la cual muestra cómo se corresponden en un sentido cualitativo (la flecha indica que una propiedad afecta positivamente a un atributo, es decir, mejora este atributo). El primer paso consiste en relacionar los atributos de usabilidad de la ISO/IEC 25010:2011 con las propiedades de usabilidad específicas que determinan las características de usabilidad que han de ser mejoradas en el sistema. El segundo paso se plantea para identificar patrones de usabilidad que pueden incorporarse a la arquitectura de software para mejorar la usabilidad del sistema final, abordando una necesidad indicada por una propiedad de usabilidad. Es importante destacar, que los patrones de usabilidad pueden abordar una o varias propiedades de usabilidad, y las propiedades de usabilidad pueden mejorar uno o varios atributos de usabilidad.

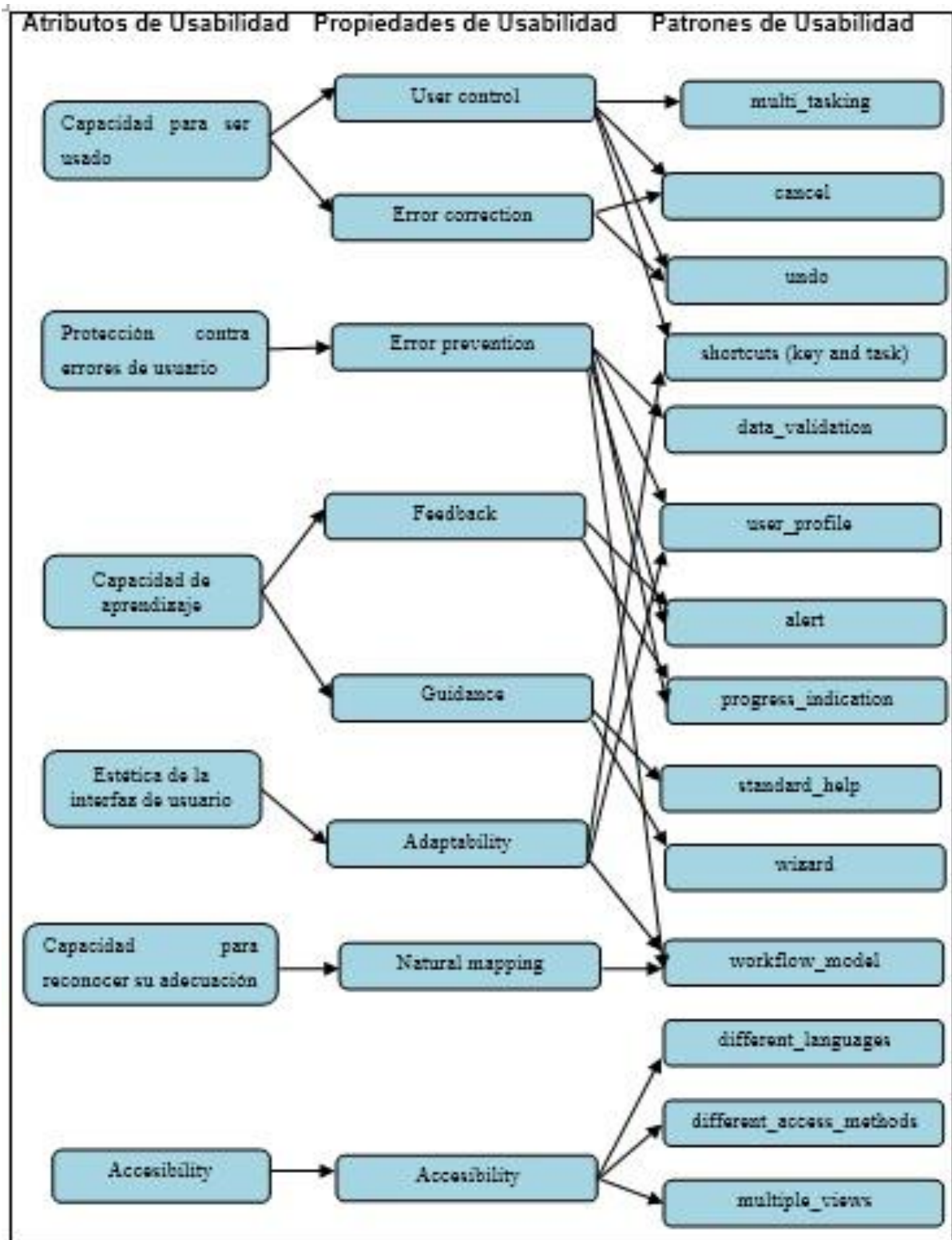


Figura 12: Relación entre atributos, propiedades y patrones de usabilidad.

Fuente: Modificado de (Folmer, Gulp et al. 2003).

Con el desarrollo de los requisitos y patrones de usabilidad se brinda una clara visibilidad de los aspectos de usabilidad a considerar en el desarrollo de un producto. Para realizar las actividades con la calidad requerida en cada una de las etapas de la Ingeniería de la Usabilidad, es necesario

la intervención de ciertos roles, evitando de esta forma, un impacto negativo en la duración y costos del producto. Seguidamente se abordará sobre las responsabilidades que poseen cada uno de ellos en la usabilidad.

2.5 Roles y responsabilidades que intervienen en la Ingeniería de Usabilidad

Un proceso necesita recursos humanos y materiales para su ejecución. Es necesario el compromiso de los miembros del proyecto para llevar a cabo todas las actividades que se definan. Pero no sólo el compromiso es importante, también cada miembro del proyecto debe tener bien definido el rol a cumplir y las responsabilidades a asumir (Ramos Blanco 2013).

En el programa de Mejoras, llevado a cabo con la evaluación del nivel 2 de CMMI en 3 de los centros de desarrollo de la UCI, se describen roles y responsabilidades a tener presente en la administración de requisitos de manera en general, así como en el Modelo de Calidad para Desarrollo de Aplicaciones Informáticas (MCDAI), en construcción actualmente. Cabe destacar que a pesar de definirse estos 2 modelos, en la categoría de ingeniería, aún no se cuenta con un proceso que encauce las buenas prácticas de la Ingeniería de Usabilidad y contribuya al éxito esperado. Se presenta como deficiencia la no identificación del rol diseñador, actor primordial para desarrollar actividades necesarias en la usabilidad y responsabilidades fundamentales de otros de los roles como el del arquitecto de información (rol no valorado en la primera versión del MCDAI). A continuación se definen los roles involucrados (Ver Figura 13) y sus responsabilidades en el proceso de desarrollo de la Ingeniería de Usabilidad, partiendo de sus conocimientos en ingeniería de software, y en las tecnologías utilizadas en el desarrollo del producto.

Administrador de la calidad

- Participa en la validación de los requisitos de usabilidad, apoyándose en la lista de chequeo definida por el proyecto.
- Registra las No Conformidades (NC) en el Registro de NC manteniendo este actualizado.

Analista de Software

- Caracteriza a los clientes acorde a criterios establecidos, identificando los que serán proveedores válidos de requisitos.
- Interactúa con el cliente en la definición de los requisitos de usabilidad para identificar cuáles formarán parte de la lista de los requisitos de usabilidad del sistema.
- Define los requisitos de usabilidad a partir del análisis de los requisitos del cliente, como resultado de la fase de elicitación de los requisitos.
- Participa en el análisis y especificación de los requisitos de usabilidad (ERU), garantizando que cumpla adecuadamente con las características de los requisitos.
- Se compromete a la realización de los requisitos de usabilidad y le da seguimiento durante todo el desarrollo del proyecto.

- Participa en la validación de los requisitos de usabilidad, apoyándose en la lista de chequeo definida por el proyecto.

Arquitecto de Información

- Identifica la visión, misión y objetivos del producto, equilibrando las necesidades de la organización patrocinadora y la de sus usuarios.
- Realiza el estudio de homólogos para conocer el estado del arte del producto que se quiere desarrollar.
- Realiza auditoría de información identificando las entidades de recursos de información conociéndose como: servicios, fuentes, sistema, contenidos.
- Realiza la organización y representación de los contenidos y servicios de la página o pantalla a través de definición de la taxonomía.
- Define el sistema de etiquetado de la aplicación.
- Define los elementos del sistema de navegación de la aplicación.
- Participa en el análisis y especificación de los requisitos de usabilidad (ERU), garantizando que cumpla adecuadamente los requisitos con las subcaracterísticas de Usabilidad.
- Participa en la validación de los requisitos de usabilidad, apoyándose en la lista de chequeo definida por el proyecto.

Diseñador

- Interactúa con el cliente en la definición de los requisitos de usabilidad.
- Construye el prototipo de la interfaz de usuario luego de las especificaciones dadas por el arquitecto de información.
- Define las pautas de diseño: cromática (color), iconográfica, tipográfica, apoyo gráfico, resoluciones de pantalla, grillas (banner, menú, imágenes y pie de página o pantalla), que deben ser utilizadas en la aplicación.
- Diseña y provee el logotipo e isotipo de la aplicación.
- Crea elementos visuales atractivos que transmitan una idea específica.
- Diseña y realiza digitalmente las imágenes necesarias.
- Realiza la forma visual de la interfaz de usuario, esquema de pantallas, modelos de interfaz gráfica.
- Participa en el análisis y especificación de los requisitos de usabilidad (ERU), garantizando que cumpla adecuadamente los requisitos con las subcaracterísticas de Usabilidad.
- Participa en la validación de los requisitos de usabilidad, apoyándose en la lista de chequeo definida por el proyecto.

Jefe de proyecto

- Participa en la fase de estudio preliminar (visión general del proyecto, análisis de factibilidad, proyecto técnico).

- Coordina, organiza y controla todas las tareas que se asignan a los miembros del equipo de desarrollo.
- Participa en la aceptación de los requisitos de usabilidad.
- Asigna responsables a las NC detectadas para su resolución y monitorea el cierre de estas tareas.

Cliente

- Entrega a los miembros del proyecto un catálogo con el personal de su organización.
- Imparte capacitación sobre el negocio a informatizar durante todas las etapas del desarrollo, en caso de necesitar conocimiento técnico muy complejo; realizando la función de especialista funcional.
- Participa en los encuentros coordinados por los miembros del proyecto brindando información.
- Participa en la aceptación de los requisitos de usabilidad.

Arquitecto de software

- Define todos los elementos bases de la arquitectura del proyecto.
- Identifica todos los posibles escenarios de despliegue de la aplicación.
- Identifica componentes horizontales de la aplicación.
- Determina de conjunto con los diseñadores las interfaces de integración, tanto internas como externas.
- Elabora el documento de arquitectura de software.
- Define las herramientas, bibliotecas, componentes, frameworks y otros componentes que permitan acelerar y mejorar el trabajo del proyecto.
- Define de conjunto con el jefe de proyecto, el flujo de desarrollo basado en las herramientas identificadas.
- Vela por el cumplimiento de los requisitos de hardware.
- Responsable de la integración de los componentes del sistema.



Figura 13: Roles involucrados en el proceso de desarrollo de la usabilidad.

Fuente: Elaboración propia.

Conclusiones

En este capítulo se propuso como incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software, apoyándose en la utilización de los requisitos y patrones de usabilidad definidos. La relación de los patrones con los atributos y propiedades de usabilidad, ayudaron a corregir y disminuir los problemas de usabilidad desde etapas tempranas de desarrollo del software, evitando un sobre diseño y tiempo de producción en las aplicaciones informáticas. Los roles definidos como el diseñador y el arquitecto de información contribuyeron al éxito en la realización de actividades fundamentales de la Ingeniería de Usabilidad.

CAPÍTULO III: VALIDACIÓN DEL PROCEDIMIENTO

Introducción

En el presente capítulo se validan los resultados de la investigación a través del uso de métodos cuantitativos y cualitativos como: entrevista a profundidad, grupo focales, consulta a expertos, Técnica de ladov y cuasi experimento con postprueba únicamente y grupos intactos. Se explica la forma en que son aplicados los métodos, teniendo en cuenta el propósito de la validación y con todos ellos se realiza una triangulación metodológica para lograr una mayor precisión y objetividad de las comprobaciones.

3.1 Consulta a expertos

La consulta a expertos consiste en buscar los criterios de varios especialistas en el tema, con el propósito de validar y disponer de una base de apoyo que respalde la solución a utilizar dándole así un carácter científico al tema de investigación, ya que los niveles de conocimientos que se involucran en estos métodos son muy particulares y específicos. A los resultados obtenidos se le dará un tratamiento estadístico. Se empleó como apoyo para el análisis estadístico el excel, reuniendo la mayor cantidad posible de elementos que aportaron datos significativos.

3.1.1 Selección de expertos

Existen diversos criterios sobre la cantidad de expertos a seleccionar. Astigarraga menciona que aunque no hay forma de determinar el número óptimo de expertos, no es aconsejable recurrir a más de 30 expertos por lo engorroso que resultaría el análisis estadístico (Astigarraga, Dow et al. 2010). Un elemento significativo está dado por las preguntas de la encuesta realizada, la cual debe ser clara, precisa y sin ambigüedades, ajustándose al objetivo trazado.

De una total de 18 expertos se seleccionaron por su coeficiente de competencia 14. Los escogidos tienen amplios conocimiento en el área de ingeniería de requisitos, normas internacionales y en calidad de software. El 7 % es Doctor en Ciencias, el 80 % es Máster y el 13% es Ingeniero en Ciencias Informáticas, estos últimos fueron seleccionados por tener basto dominio en el desarrollo de investigaciones asociadas a la Ingeniería de Software y Calidad de requisitos. El 85% de los encuestados son asistentes, el 13% profesores auxiliares y el 2 % profesores titulares, los cuales tienen más de 5 años de experiencias trabajando en disímiles proyectos productivos. El 90% son tutores de tesis de pregrado y maestría relacionadas con la usabilidad. El 70% ha desempeñado el rol de analista principal por varios años en proyectos de exportación. El 82.5% se ha desempeñado en roles como: jefe de proyecto, administrador de la calidad, analista, diseñador, en proyectos de alta importancia.

A partir de estos datos se calcula el coeficiente de competencia (K) para lo que se tiene en cuenta la autovaloración del experto acerca de su competencia (Kc coeficiente de conocimiento) y las fuentes de argumentación (Ka coeficiente de argumentación) mediante la siguiente fórmula (1):

$$K = \frac{Kc + Ka}{2} \quad (1)$$

El Ka se calcula a partir de una escala predeterminada (Ver Tabla 7) que se incorpora a la encuesta aplicada a los expertos.

Tabla 7: Escala de puntuación para calcular coeficiente de argumentación.

Fuente: Elaboración propia.

Fuentes de argumentación	Alto	Medio	Bajo
Grado científico	0.30	0.20	0.10
Publicaciones nacionales según su nivel	0.05	0.05	0.05
Publicaciones internacionales según su nivel	0.05	0.05	0.05
Experiencia en el tema	0.50	0.40	0.20
Tutor de trabajos investigativos	0.30	0.20	0.10
Intuición según su aval	0.10	0.10	0.10

Se le dio un rango de más valor a: experiencia en el tema, grado científico y tutor de trabajos investigativos por ser variables de más pesos. La encuesta presentada se encuentra en el Anexo 3. A partir de estos datos se especifica el grado de coeficiente de competencia.

Si $0,8 \leq K < 1,0$ coeficiente de competencia Alto.

Si $0,5 \leq K < 0,8$ coeficiente de competencia Medio.

Si $K < 0,5$ coeficiente de competencia Bajo.

En la Tabla 8 se contiene un resumen de los grados y coeficientes de competencia, coeficiente de conocimiento y coeficiente de argumentación de cada experto, para determinar estos valores se estableció un rango entre 0 y 1 de acuerdo al currículo de los especialistas.

Tabla 8: Grado y coeficiente de competencia, coeficiente de conocimiento y argumentación de los expertos.

Fuente: Elaboración propia.

No. Experto	K _c	K _a	K	Grado
1	0.70	0.80	0.75	medio
2	0.80	0.80	0.80	alto
3	0.60	0.70	0.65	medio
4	0.90	0.80	0.85	alto
5	0.80	0.90	0.85	alto
6	0.60	0.70	0.65	medio
7	0.70	0.90	0.80	alto
8	0.70	0.80	0.75	medio
9	0.80	0.70	0.75	medio
10	0.70	0.70	0.70	medio
11	0.60	0.80	0.70	medio
12	0.70	0.70	0.70	medio
13	0.80	0.80	0.80	alto
14	0.80	0.90	0.85	alto

A partir de esto se determinó que los expertos para validar la propuesta poseen un 42.9% grado de coeficiente de competencia alto y un 57.1% un grado medio, confirmándose que la mayoría de los expertos seleccionados tienen un grado de coeficiente de competencia medio.

3.1.2 Escalamiento de Likert

Con el fin de analizar la opinión de los expertos sobre la solución propuesta en la investigación, se aplica la escala de Likert, para evaluar el grado de concordancia con las ideas planteadas en función de valorar el cumplimiento de la hipótesis de la presente investigación. En esta escala se otorga una puntuación entre 1 y 5 a cada ítem permitiendo a los entrevistados emitir una valoración (Ver anexo 4). Luego se calculó un índice porcentual (IP) según la fórmula (2), que integra en un solo valor la aceptación del grupo de evaluadores a partir de sus respuestas.

$$IP = \frac{5 (\% \text{ de } MA) + 4 (\% \text{ de } A) + 3 (\% \text{ de } AA) + 2 (\% \text{ de } PA) + 1 (\% \text{ de } I)}{5} \quad (2)$$

Donde:

- MA: Muy adecuada (Muy importante, En todos los casos)
- A: Adecuada (Importante, En la mayoría de los casos)
- AA: Algo adecuada (Algo importante, En algunos casos)

- PA: Poco Adecuada (Poco Importante, En la minoría de los casos)
- I: Inadecuada (Nada importante, En ningún caso)

En la Figura 14 se muestra el resultado del procesamiento realizado a través del escalamiento de Likert, la cual evidencia que el IP en todos los casos supera el 80%. Por otra parte, muestra que la solución es valorada en un alto grado como muy adecuada, para lograr el cumplimiento de las actividades y artefactos en la gestión de los requisitos de usabilidad, la definición de los mismos para guiarse en los proyectos productivos, así como los roles y responsabilidades descritos. El criterio de mayor índice porcentual de coincidencia fue la valoración sobre los requisitos de usabilidad definidos, sobrepasando el 95%. Esto demuestra que los expertos coinciden en considerar que con la solución propuesta, se logra gestionar satisfactoriamente los requisitos de usabilidad, así como incorporar patrones de usabilidad al proceso de desarrollo de un producto de software, lo cual constituye un aporte importante de la investigación a la solución del problema inicialmente planteado.

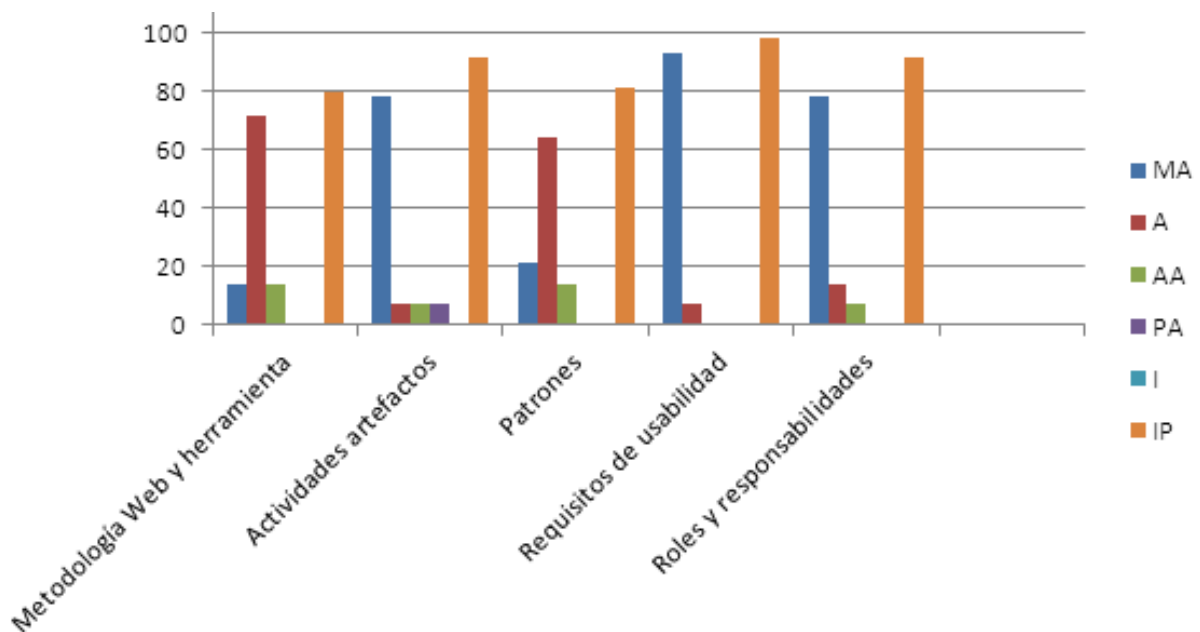


Figura 14: Resultados de la aplicación de la escala de Likert.

Fuente: Elaboración propia.

3.2 Técnica de ladov

La técnica de ladov en su versión original fue creada por N. V. Kuzmina para el estudio de la satisfacción por la profesión en carreras pedagógicas (Kuzmina 1970). Posteriormente ha sido utilizada por diferentes investigadores para evaluar la satisfacción de determinados sujetos respecto a un elemento específico (Ramos Blanco 2013).

Esta técnica constituye una vía para el estudio del grado de satisfacción de los implicados. Se basa en el análisis de un cuestionario que tiene una estructura interna determinada, la cual se

fundamenta en las relaciones que se establecen entre tres cerradas (cuya relación la persona desconoce) y dos preguntas abiertas para un análisis posterior. Las preguntas cerradas fueron elaboradas por la autora y se establecen a través del denominado "Cuadro Lógico de ladov" (Ver Tabla 9), indicando el número resultante de la interrelación la posición de cada sujeto en la escala de satisfacción. La misma es la siguiente:

- Clara satisfacción.
- Más satisfecho que insatisfecho.
- No definida.
- Más insatisfecho que satisfecho.
- Clara insatisfacción.
- Contradictoria.

Tabla 9: Cuadro Lógico de ladov.

Fuente: Modificado por la autora.

	1. ¿Considera que la incorporación de la Ingeniería de Usabilidad en el proceso de desarrollo del software es complejo y difícil de entender?								
	No			No sé			Si		
3. ¿Le satisface sus expectativas la solución propuesta?	2. ¿Si Ud. fuera a realizar otro proyecto utilizaría los requisitos de usabilidad propuestos así como los patrones de usabilidad?								
	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me es indiferente	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta	6	6	6	6	4	4	6	4	5
No puedo decir	2	3	6	3	3	3	6	3	4

Para medir el grado de satisfacción de los proyectos se tomó una muestra de 24 usuarios con diferentes roles en diversos proyectos: 11 analistas de software, 2 arquitectos de información, 4 jefes de proyecto, 3 arquitectos de software y 4 administradores de la calidad, a los cuales se les aplicó una encuesta (Ver Anexo 5). Utilizando el Cuadro Lógico de ladov definido anteriormente se recopilaron los resultados obtenidos de cada encuestado y se fue tomando la escala de satisfacción de cada resultado obtenido. El resultado del nivel de satisfacción según los encuestados se muestra en la Figura 15.



Figura 15: Resultado de la aplicación de la técnica de ladov y valor del Índice de Satisfacción Grupal (ISG).

Fuente: Elaboración propia.

El ISG oscila entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción.

ladov contempla además dos preguntas complementarias de carácter abierto. Estas permiten conocer las causas de los niveles de satisfacción obtenidos. A las mismas respondieron:

- Se considera útil la solución propuesta, pues permite obtener una visión más clara de la Ingeniería de Usabilidad y por ende eleva sus resultados en los productos finales.
- Resaltan la necesidad de la existencia de una guía que ayude al analista a desarrollar los requisitos de usabilidad en los mismos con la calidad requerida.
- Se destaca como elemento positivo la generalidad y flexibilidad de la solución propuesta, pues permite adecuarla a cualquier proyecto productivo.
- Se manifiesta el beneficio de incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software a través de la gestión de los requisitos y patrones de usabilidad.

- Coinciden de muy importante, la inclusión del diseñador al trabajo conjunto del analista y del arquitecto de información para el éxito en el desarrollo de los requisitos de usabilidad.
- Se señala positivamente la facilidad que brinda la solución para ponerla en práctica en los proyectos sin desmotivar a los usuarios por el volumen de información que se genere.
- Recomiendan implantar en varios centros o instituciones donde se desarrollan productos de software la solución que se propone.

La aplicación de la técnica de ladov ha permitido confirmar la satisfacción de los involucrados respecto a la implantación de la solución. Las recomendaciones han de ser consideradas para introducir mejoras a la propuesta.

3.3 Grupos focales

El grupo focal se emplea como técnica de investigación cualitativa. Este “es un tipo de entrevista grupal que se estructura para recolectar opiniones detalladas y conocimientos acerca de un tema particular, vertidos por los participantes seleccionados” ((Balcázar, González et al. 2005),(Cañizares 2012)).

El objetivo principal de este método es validar la propuesta de solución de la investigación emitiendo criterios sobre los aspectos tratados. Es importante destacar, que en esta actividad el investigador se comporte de manera reservada, tomando nota de las ideas expresadas por los participantes para no influir en las opiniones dadas por los integrantes del grupo.

Un beneficio que aporta este método a las investigaciones está dado cuando el moderador estimula a los participantes a emitir ideas sobre el asunto que se estudia. La interacción entre los mismos dará lugar a la consideración de interesantes aspectos adicionales o identificará problemas comunes experimentados por muchas personas (Cañizares 2012).

Se seleccionaron 7 participantes para desarrollar esta actividad, entre ellos: 2 especialistas del grupo de Ingeniería de Prueba, en particular de usabilidad, del DEPSW de CALISOFT, 1 especialista de la Vicerrectoría de Producción con bastas experiencias en esta rama, 1 diseñador, 1 arquitecto de software, 1 arquitecto de información y 1 analista de software. El 57.1% de los implicados presentan la categoría científica de máster y han trabajado en proyectos de exportación. No han intervenido de manera directa 2 de los involucrados, debido a que no está presente algunos de los roles propuestos como el diseñador y el arquitecto de información, en la estructura de distintos proyectos.

Antes de comenzar el debate se realizó una presentación de los resultados obtenidos al incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software. Para guiar esta actividad se tuvo presente ir de lo general a lo específico, de lo más fácil a lo más difícil, propiciando estructuradamente el debate en base a las experiencias personales y al conocimiento que poseen sobre la temática cada uno de los presentes. Las preguntas realizadas se pueden observar en el Anexo 6.

Durante las intervenciones se pudo concluir que todos estaban de acuerdo con la necesidad de introducir los procesos de desarrollar los requisitos y patrones de usabilidad, donde participaran los roles antes mencionado como primordial en la realización de esta tarea. Además, se pudo manifestar la conformidad de que cada proyecto adoptara la solución propuesta, adaptándola de manera flexible en los proyectos productivos para disminuir la cantidad de NC de usabilidad en los productos finales.

Se realizó la sugerencia de implementar en varios centros o instituciones donde se desarrollan productos de software la solución. Esta propuesta, al igual que en la Técnica de ladov, se contempla en las recomendaciones de la investigación.

3.4 Entrevista a profundidad

Para la validación de la propuesta se utiliza el instrumento cualitativo entrevista a profundidad, con el objetivo de seleccionar los especialistas que intervendrán con su conocimiento en validar la solución propuesta. Este método es la entrevista profesional que se realiza entre un entrevistador y un entrevistado con el objeto de obtener información sobre la vida, tema, proceso o experiencias concreta de una persona. A través de la entrevista a profundidad, el entrevistador conoce lo que es importante y significativo para el entrevistado; comprende cómo ve, clasifica e interpreta su mundo en general o tema que interesa para la investigación, en particular (Ruiz 1996).

Tiene el objetivo de obtener información difícil de observar y sobre la cual el entrevistado posee conocimientos, experiencia, datos relevantes y una interpretación del fenómeno a tratar, que en otras condiciones no estaría dispuesto a compartir. Es un proceso rápido que busca .disminuir la indeterminación y redundancia de los procesos de investigación ((Brito Riverol 2014), (Valles 2003)). En base a las hipótesis y teorías de partida, el entrevistador debe llevar escritas en un guión las preguntas o temas que considera interesante analizar, señalando el tiempo que dedicará a cada una para que el entrevistado se sienta confiado con la entrevista desde un principio.

Los entrevistados fueron especialistas en calidad e ingeniería de software, con conocimientos y experiencias en temas de requisitos, arquitectura de software y de la característica usabilidad, pertenecientes a las siguientes instituciones:

- Universidad de las Ciencias Informáticas, departamento de Calidad de Software.
- Universidad de las Ciencias Informáticas, centro productivo.
- DESOFT.
- DEPSW de CALISOFT.

Se realizó una breve exposición sobre la propuesta por parte de la autora y seguidamente, se ejecutaron las preguntas definidas como hilo conductor de la entrevista (Ver Anexo 7). Los entrevistados estuvieron de acuerdo con la propuesta definida y expresaron su conformidad en cuanto a la posibilidad de que cada proyecto adoptara la misma, con el objetivo de elevar la calidad del proceso de evaluación de los productos de software al disminuir la cantidad de NC

detectadas de usabilidad. Además, propusieron extender la solución a los restantes proyectos en desarrollo del país.

3.5 Cuasi experimento con postprueba únicamente y grupos intactos para medir la variable nivel de disminución de la cantidad de NC de usabilidad

Los diseños cuasi experimentales manipulan deliberadamente al menos una variable independiente para ver su efecto y relación con una o más variables dependientes. Un cuasi experimento se utiliza cuando no es posible la realización de un experimento verdadero, o sea, cuando los grupos que recibirán el tratamiento experimental, no están asignados aleatoriamente, sino que están conformados antes del experimento (grupos intactos) (Sampieri and Collado 2008). Los tipos de cuasi experimentos son: diseño con postprueba únicamente y grupos intactos, diseño con preprueba-postprueba y grupos intactos y por último diseño cuasi experimental de series cronológicas (Sampieri and Collado 2008).

Diseño con postprueba únicamente y grupos intactos: este diseño utiliza a dos grupos, uno recibe el tratamiento experimental y el otro no. Los grupos son comparados en la postprueba, para analizar si el tratamiento experimental tuvo un efecto sobre la variable dependiente.

Diseño con preprueba - postprueba y grupos intactos: este diseño es similar al de postprueba únicamente y grupos intactos, solamente que a los grupos se les administra una prueba. La cual puede servir para verificar la equivalencia inicial de los grupos.

3. Diseño cuasi experimental de series cronológicas: este diseño se emplea cuando el investigador desea analizar efectos al mediano y largo plazo o efectos de administrar varias veces el tratamiento experimental, y no cuenta con la posibilidad de asignar al azar a los sujetos a los grupos del experimento. En este caso, pueden utilizarse los diseños cuasi experimentales salvo que los grupos son intactos.

En esta investigación se emplea el diseño con postprueba únicamente y grupos intactos, para comprobar la validez de la variable dependiente: nivel de disminución de la cantidad de NC de usabilidad, obtenida a partir de las evaluaciones de usabilidad en el Departamento de Evaluación de Productos de CALISOFT.

El primer grupo estuvo compuesto por cuatro productos de software evaluados por el DEPSW de CALISOFT sin tener en cuenta la propuesta de solución, y el segundo grupo lo formó estos mismos productos de software pero con el procedimiento propuesto.

El propósito de la investigación realizada es disminuir la cantidad de NC de usabilidad detectadas en el proceso. Para demostrar la validez de esta variable dependiente en la evaluación a productos de software, se establece un análisis de las NC de usabilidad detectadas antes y después de la implantación de la propuesta. En la Tabla 10 se muestran los resultados de la evaluación de la usabilidad en un producto de software del segundo grupo, aplicándoles los

patrones de usabilidad *data_validation*, *standard_help*, *progress_indication* y *undo*; y/o requisitos de usabilidad (RU) asociados a las subcaracterísticas de usabilidad de la ISO/IEC 25010: 2011.

Tabla 10: Descripción de NC de un producto de software antes y después de la propuesta.

Fuente: Elaboración propia.

No	Descripción de NC	Clasificación	Se elimina la NC aplicando la propuesta
1	En el campo de selección “Activa”, aparece escrita incorrectamente la palabra “Sí”.	Ortografía	
2	Cuando no se introduce criterio de búsqueda alguno, y se presiona el botón “Buscar”, el sistema muestra un mensaje que no es el más sugerente.	Calidad del contenido	RU
3	El listado de las categorías y dominios no tienen un orden específico.	Navegabilidad	RU
4	Los campos de introducción de textos de la funcionalidad “Dominio” permiten introducir grandes cantidades de caracteres que no están preparados para procesar, distorsionan la interfaz, pues al listar los dominios no realizan el salto de línea.	Error de Interfaz	Patrón <i>data_validation</i>
5	Los campos de introducción de textos de la funcionalidad “Categoría” permiten introducir grandes cantidades de caracteres que no están preparados para procesar, distorsionan la interfaz, pues al listar las categorías no realizan el salto de línea.	Error de Interfaz	Patrón <i>data_validation</i>
6	El paginado no permite ir directamente ni a la primera ni a la última página del listado.	Control del Usuario	RU
7	Al introducir un nombre igual a uno ya existente, el sistema muestra un mensaje poco sugerente con respecto a	Formato	Patrón <i>standard_help</i> y RU

	lo que ocurre. Además no se mantiene la uniformidad en cuanto a los otros mensajes de “nombres repetidos” restantes.		
8	Cuando se muestra un mensaje de error los nombres de los “campos” no se mantienen al lado de su “campo de texto” correspondiente.	Error de Interfaz	
9	El mensaje de confirmación que se muestra al “Eliminar”, no tiene el formato de la página, y la información que brinda no es suficiente.	Formato	RU
10	El sistema muestra 3 tipos de mensajes (Información, Éxito y Error) y solo debe tener dos tipos, los cuales son de información y/o de error. Deben estandarizar la posición donde se mostrará el mensaje.	Formato	
11	Al editar un “usuario” y “guardar”, la interfaz que se recomienda que salga seguidamente es la del “listar usuarios” y no la de la edición.	Recomendación	
12	El pie de página no tiene el año actualizado.	Calidad del contenido	RU
13	Al realizar la búsqueda de imágenes y obtener los resultados, la imagen seleccionada por el cursor debía resaltar con respecto a las otras no seleccionadas. “Evento <i>mouseover</i> ”.	Visibilidad	RU
14	En el menú de Administración cuando se visita un enlace determinado, éste no se diferencia del resto de los enlaces restantes.	Navegabilidad	RU
15	Cuando se comete un error al introducir datos en el formulario, debe señalarse	Error Interfaz	Patrón <i>data_validation</i>

	en un color que resalte los campos incorrectos de alguna forma.		
16	Los enlaces rotos que la búsqueda arroja como resultado una vez seleccionados no se guardan como consultados con respecto a los otros enlaces que no han sido seleccionados.	Navegabilidad	RU
17	Cuando se abre el buscador con Internet Explorer aparecen unos cuadros al lado de los íconos.	Compatibilidad	RU
18	No existe un cambio visible cuando se pasa el cursor por encima de los enlaces (no se contempla el cambio del cursor).	Visibilidad	RU
19	La fuente utilizada para el texto que se introduce en el campo de búsqueda tiene muy bajo el puntaje.	Legibilidad	
20	La tipografía utilizada para el texto que se introduce en el campo de búsqueda tiene un color que no es adecuado.	Visibilidad	RU
21	No es necesario tener texto aclaratorio dentro del campo de búsqueda que indique la acción de buscar. Esto quita estética.	Error de interfaz	
22	El contorno del campo de búsqueda debe ser lo más neutral posible (el naranja utilizado en determinados monitores se ve rojo y da la idea de error).	Error de interfaz	
23	La iconografía no tiene buen rendimiento, están pixelados en sus bordes. Cuando se abre el buscador con el navegador "Safari" es cuando más acentuado se ve el problema.	Error de interfaz	

24	El color naranja de los botones está demasiado saturado.	Recomendación	
25	La búsqueda avanzada no se muestra de la misma forma cuando se accede desde la página de inicio o de la de resultados de la búsqueda.	Recomendación	
26	La página de error no está diseñada con el aspecto general del sitio, debe mostrar el logo y el nombre del sitio. No debe mostrar una página de "Error 404"	Error de interfaz	RU
27	El botón de la búsqueda avanzada no tiene una buena proporción entre tamaño del texto y tamaño del botón.	Error de interfaz	
28	Se realizó alguna búsqueda que no produjo resultados esperados y que se conoce que hay sitio relacionados con el tema. Ej. Específico red de cubana de ciencia.	Calidad del contenido	RU
29	Las opciones de búsqueda avanzada no están disponibles para todas las categorías de búsqueda definidas.	Control del usuario	
30	El buscador no ofrece información sobre la entidad responsable del sistema.	Recomendación	RU
31	No se ofrece información de contacto.	Recomendación	RU
32	La búsqueda de imágenes es demasiado básica.	Control del usuario	
33	El bloque de las imágenes que se muestra en la página de resultados de la búsqueda no se encuentra alineado a la izquierda del campo de búsqueda como el bloque de los enlaces en la página de resultados de la web.	Error de interfaz	
34	Cuando se visita una categoría aparece al lado del logo un texto con el nombre de la categoría.	Error de interfaz	

35	El objetivo del buscador es precisamente ese “Buscar”, así que todas las ventajas y apoyo que se le puedan dar al usuario para auxiliarlo cuenta. Se recomienda implementar la sugerencia por términos.	Control del usuario	del Patrón <i>standard_help</i> y patrón <i>progress_indication</i>
36	El mensaje de error no permite volver a la situación anterior.	Control del usuario	del Patrón <i>undo</i> y RU
37	El buscador no ofrece alguna forma de retroalimentación para con los usuarios Ej.: formulario de contacto para reportar errores, sugerencias, etc.	Recomendación	RU
38	En la ventana principal del buscador para introducir información, el cursor no es ubicado en donde el dato debe ser introducido.	Control del usuario	del RU
39	Los textos aclaratorios de la búsqueda avanzada tienen un puntaje muy bajo y el color es muy claro.	Legibilidad	RU
40	La zonificación establecida es extremadamente grande lo cual atenta contra la legibilidad, usabilidad y la ergonomía.	Error de interfaz	
41	El logo no tiene un color identificativo.	Calidad del contenido	del RU
42	El buscador no cuenta con una ayuda que contribuya a la efectividad de la búsqueda que pueden realizar usuarios de poca experticia.	Control del usuario	del Patrón <i>standard_help</i>
43	La jerarquía tipográfica es deficiente.	Error de interfaz	RU
44	Los campos de la búsqueda avanzada son demasiado estrechos.	Error de interfaz	
45	Los campos de la búsqueda avanzada no están situados a la misma distancia, ni tienen el mismo tamaño.	Error de interfaz	

46	El campo de búsqueda es demasiado grande.	Error de interfaz	
47	Es necesario dar <i>scroll</i> para ver el pie de página en la página principal.	Error de interfaz	RU
48	El posicionamiento de los enlaces a páginas externas en la parte superior izquierda utilizando iconos sin un título que identifique que son, dan la idea de ser funcionalidades del buscador.	Control del usuario	RU
49	Los íconos utilizados para redireccionar a un conjunto de sitios utilizan logos con los cuales el usuario no está familiarizado, ni conoce, ya que son productos nuevos y algunos de ellos aún no están disponibles.	Control del usuario	
50	El lenguaje está más cercano al informático o técnico que al del usuario relacionado con los operadores que se muestran en el campo de búsqueda cuando se realiza una búsqueda avanzada.	Visibilidad	RU
51	No existe un cambio visible cuando se pasa el cursor por encima de los enlaces (no se contempla el cambio del cursor).	Error de interfaz	RU
52	Existen enlaces rotos.	Navegabilidad	RU

Luego de aplicada la propuesta a los productos de software sometidos a esta evaluación, se observa una disminución significativa en la cantidad de NC de usabilidad en el proceso de evaluación de usabilidad en los 4 productos de software. Ver tabla 11.

Tabla 11: Cantidad de NC de productos antes y después de la propuesta.

Fuente: Elaboración propia.

Productos de software	Cantidad de NC antes de la propuesta	Cantidad de NC después de la propuesta	% de disminución de cantidad de NC de usabilidad
A	52	20	61,5 %
B	34	16	52,9 %
C	48	21	56,3 %
D	47	18	61,7 %

Al observar la tabla 11 se puede decir que antes de aplicar el procedimiento solamente los proyectos definían elementos de usabilidad a nivel de diseño de interfaz, y en la definición de los requisitos de usabilidad se referían a elementos vagos e imprecisos o ese aspecto no lo consideraban; de lo que se deriva que el valor de la variable independiente: nivel de incorporación de la Ingeniería de Usabilidad en el proceso de desarrollo del software, es bajo. Con el procedimiento propuesto donde se describen actividades, artefactos, roles y responsabilidades que integra la Ingeniería de Usabilidad en el proceso de desarrollo del software, se puede plantear que la variable independiente obtiene un valor alto. Al aplicar el mismo se observa una disminución significativa de las NC de usabilidad, ya que sobrepasa el 50 % en todos los productos de software analizados. Lo que concluya que con un alto de nivel de incorporación de la Ingeniería de Usabilidad en el proceso de desarrollo del software se obtiene un alto nivel de disminución de la cantidad de NC de usabilidad.

3.6 Triangulación metodológica

La triangulación metodológica es un término para el análisis de datos recopilados a través de diferentes métodos, lo cual permite analizar una situación desde diversos ángulos. Es un procedimiento de control implementado para garantizar la confiabilidad en los resultados de cualquier investigación, debido a que compara resultados obtenidos mediante métodos cuantitativos y cualitativos teniendo en cuenta que las debilidades de cada método individual van a ser compensadas por la fortaleza contra balanceadora del otro (Cañizares 2012).

La definición de triangulación ofrecida por (Denzin 1990) y con la cual la autora de la investigación coincide dice que es, la aplicación y combinación de varias metodologías de la investigación en el estudio de un mismo fenómeno. La triangulación no solamente recolecta y analiza un estudio mostrando que sus conclusiones no dependen del modo utilizado para medir el grado de validez externa de los datos, sino también permite otorgar mayor confiabilidad, nivel de precisión y contrastar la consistencia interna tratada.

Se realiza la evaluación a través de la aplicación de forma simultánea de los métodos: consulta a expertos utilizando el escalamiento de Likert, grupo focal, la técnica de ladov, la entrevista a profundidad y cuasi experimento con postprueba únicamente y grupos intactos con el objetivo de validar la hipótesis planteada y la relación entre las variables independiente y dependiente (Ver Figura 16). En la consulta a expertos, teniendo en cuenta el escalamiento de Likert, la mayoría de los involucrados valoraron en un alto grado la solución propuesta, lo cual constituye un aporte importante de la investigación. Luego de realizar esta primera validación se realizó los métodos entrevista a profundidad y grupos focales a personas con vastos conocimientos en temas de usabilidad e ingeniería del software. La propuesta fue catalogada, apropiada y acorde para los fines definidos. Ello permitió proseguir de manera acertada con su implantación. Se aplicó la Técnica de ladov con el fin de conocer el nivel de satisfacción desde el punto de vista del usuario respecto a la utilización de la solución, la cual arrojó resultados positivos al alegar un alto índice de satisfacción y hacer recomendaciones de mejora. Se realizó además, un cuasi experimento con postprueba únicamente y grupos intactos, para establecer una comparación entre los resultados que se obtienen antes y después de aplicar el procedimiento observándose la validez de la variable dependiente definida en la hipótesis de la investigación.

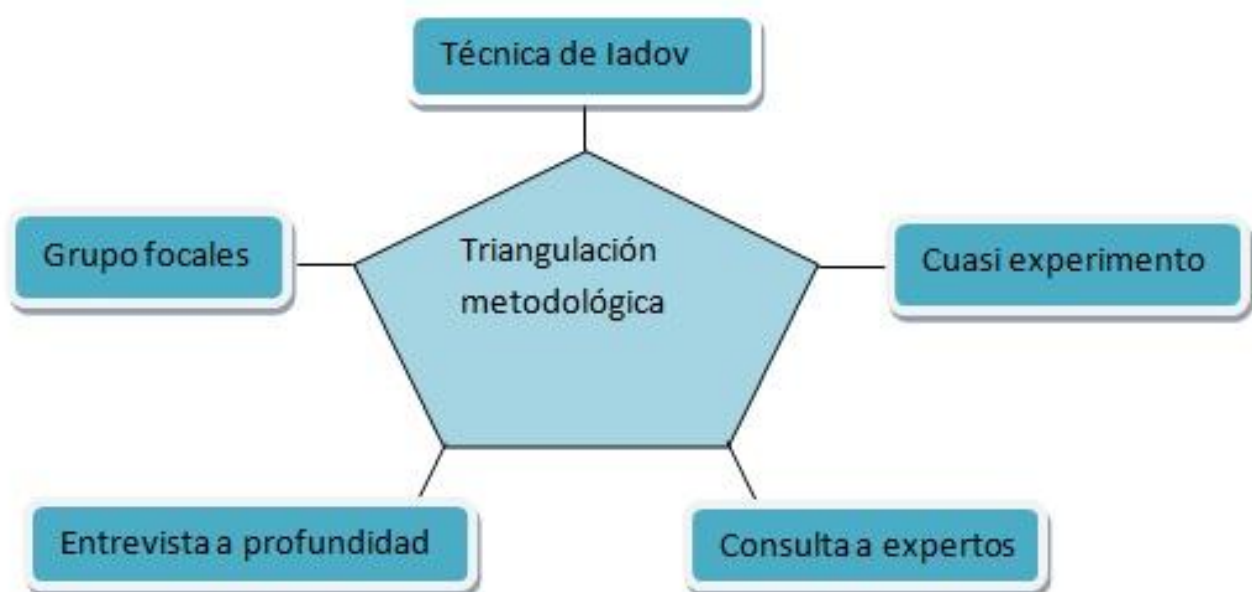


Figura 16: Métodos de validación de la propuesta.

Fuente: Elaboración propia.

Tras la aplicación y análisis de los métodos de validación antes descritos, se puede concluir que existe una correspondencia satisfactoria entre los resultados obtenidos de la variable evaluada, confirmando que la hipótesis planteada en la investigación cumplió el objetivo de disminuir la cantidad de No Conformidades de usabilidad detectadas en el proceso de evaluaciones de usabilidad en el DEPSW de CALISOFT.

Conclusiones

Después de aplicado los métodos científicos con el objetivo de validar la solución, se comprobó con el método de triangulación que existe una correspondencia positiva en los resultados obtenidos. Se logró mediante la entrevista a profundidad, los grupos focales y consulta de expertos apoyándose en el escalamiento de Likert, una valoración positiva de la solución para su implantación. A través de la aplicación de la técnica ladov se corroboró una clara satisfacción de los usuarios con la propuesta, así como con el método cuasi experimento con postprueba únicamente y grupos intactos se confirmó la validez de la variable dependiente: nivel de disminución de la cantidad de NC de usabilidad, definida en la hipótesis de la investigación.

CONCLUSIONES

La investigación realizada permitió llegar a las siguientes conclusiones:

1. Con los elementos teóricos y prácticos más actuales se desarrolló e implementó la solución para incorporar la Ingeniería de Usabilidad en el proceso de desarrollo del software, la cual contribuyó a disminuir la cantidad de NC de usabilidad en las evaluaciones realizadas en el Departamento de Evaluación de Productos de CALISOFT.
2. Los requisitos de usabilidad así como los patrones de usabilidad están en condiciones de ser introducidos en los productos de software, dado que constituye una solución flexible y fácilmente reutilizable, que cuenta con las experiencias positivas de su empleo en la UCI y DESOFT, brindando un aporte no solo teórico sino también práctico.
3. Se definió un procedimiento que incorpora la Ingeniería de Usabilidad en el proceso de desarrollo de un producto de software para disminuir la cantidad de NC de usabilidad en el proceso de evaluaciones en el Departamento de Evaluación de Producto de CALISOFT.
4. El conjunto de métodos científicos utilizados para la validación de la propuesta (grupo focal, técnica de ladov, entrevista a profundidad, cuasi experimento con postprueba únicamente y grupos intactos y consulta a expertos) permitió comprobar que la solución fue aceptada y aprobada por varios especialistas vinculados al desarrollo de aplicaciones informáticas, mostrando alta satisfacción con respecto a la necesidad, utilidad y actualidad de la solución.

RECOMENDACIONES

Las principales recomendaciones derivadas del trabajo realizado son:

1. Sistematizar la aplicación del procedimiento en varios centros o instituciones donde se desarrollan productos de software.
2. Introducir la solución en Entorno de Producción de Software Dirigido por Modelos, a través de la introducción de herramientas CASE.
3. Incorporar al procedimiento herramientas de Inteligencia Artificial que apoyen el proceso de selección de patrones de usabilidad.

REFERENCIAS BIBLIOGRÁFICAS

1. Alelú, M., S. Cantín, et al. (2011). "Estudio de Encuestas. Métodos de Investigación".
2. Astigarraga, T., E. M. Dow, et al. (2010). The emerging role of software testing in curricula. Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments.
3. Aurum, A. and C. Wohlin (2005). "Engineering and Managing Software Requirements".
4. Balcázar, P., N. González, et al. (2005). Investigación Cualitativa.
5. Bass, L. and B. John (2003). "Linking usability to software architecture patterns through general scenarios". The journal of systems and software: 187-197.
6. Brito Riverol, Y. (2014). Estrategia para estructurar las evaluaciones de un laboratorio de pruebas de software desde la perspectiva de la acreditación. La Habana, Cuba, Universidad de las Ciencias Informáticas.
7. Cañizares, R. (2012). Repositorio de recursos educativos para las instituciones de Educación Superior.
8. Capote García, T. (2011). Conceptualización e implantación de un Laboratorio Industrial de Pruebas de Software. La Habana, Cuba, Universidad de las Ciencias Informáticas.
9. Casas Anguita, J., J. Repullo Labrador, et al. (2003). "La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (I)".
10. Cervantes, H. (2010). "Arquitectura de Software".
11. Cusmai, M. (2011). Calidad de Software desde Latinoamérica. Infotesting. Un espacio para testers profesionales. Revista digital.
12. Davyt Dávila, N. (2001). Ingeniería de requerimientos. Una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto. Uruguay, Universidad de Uruguay.
13. Denzin, N. K. (1990). "Triangulation"
14. Departamento de Tecnologías de la Información (2010). "Ingeniería del Software. I.T.I Sistemas. Capítulo 2. Ingeniería de Requisitos".
15. Duran, A. and B. Bernárdez (2000). "Metodología para la Elicitación de Requisitos de Sistemas Software".
16. Escalona Cuaresma, M. J. and J. M. González Romano (2007). "Metodología y Técnicas en Proyectos software para la Web".
17. Escalona, M. J. and G. Arag (2008). "NDT. A Model-Driven Approach for Web Requirements". IEEE Trans. Softw. Eng **34**(3): 377-390.
18. Escalona, M. J. and J. M. González (2007). "Metodologías para la Ingeniería Web".
19. Escalona, M. J. and N. Koch (2004). "Ingeniería de Requisitos en Aplicaciones para la Web. Un estudio comparativo".

REFERENCIAS BIBLIOGRÁFICAS

20. Escalona, M. J. and N. Koch (2004). "Requirements Engineering for Web Applications - A Comparative Study". *Journal of Web Engineering*: 193-212.
21. Escalona, M. J., J. Torres, et al. (2003). "Desde los requisitos al modelo conceptual en NDT".
22. Escribano, A. (2010). "Ingeniería web dirigida por modelos".
23. Ferré Grau, X. (2010). "Principios Básicos de Usabilidad para Ingenieros Software".
24. Ferré, X. (2003). "Incrementos de usabilidad al proceso de desarrollo software." VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD): 293-302.
25. Florián, B. E., O. Solarte, et al. (2010). Propuesta para incorporar evaluación y pruebas de usabilidad dentro de un proceso de desarrollo de software. *Revista EIA*: 123-141.
26. Folmer, E. and J. Bosch (2004). "Architecting for usability: a survey". *Journal of Systems and Software*: 61-78.
27. Folmer, E., J. v. Gulp, et al. (2003). "A Framework for Capturing the Relationship between Usability and Software Architecture". *Software Process Improvement and Practice*: 67-87.
28. Granollers, A. (2004). MPlu+a. Una metodología que integra la ingeniería del software, la interacción persona-ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares, Universidad de Lleida.
29. Group, S. (2013). *Chaos Manifesto 2013*.
30. Guerrero Llerena, G. M. (2004). "Experiencias en la implantación de un sistema de gestión de la calidad para el proceso de producción de software".
31. Hernández Aguilar, V. (2009). *Protofase a la ingeniería de requisitos para facilitar la comprensión del negocio a informatizar en el desarrollo de software de gestión*. La Habana, Universidad de las Ciencias Informáticas.
32. Hernández Calzada, A. (2013). *Sistema de actividades para propiciar la evaluación formativa en la enseñanza de la Física*. Palma de Mallorca, España.
33. Hernández, V. and Y. Machado (2010). "Describiendo requisitos verificables". Eighth LACCEI Latin American and Caribbean Conference for Engineering and Technology. Innovation and Development for the Americas.
34. Herrera, L. J. (2003). "Ingeniería de Requerimientos, Ingeniería de Software".
35. IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology*.
36. IEEE (2006). *Standard Glossary Software Engineering Terminology*.
37. INTECO (2008). "Guía avanzada de gestión de requisitos".
38. ISO 9241-11 (1998). *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*.
39. ISO/IEC 9126-1 (2005). *Parte 1: Modelo de Calidad, Ingeniería de Software – Calidad del Producto*.

40. ISO/IEC 25010 (2011). Systems and software engineering: software product quality and system quality in use models.
41. Jokela, T., J. Koivumaa, et al. (2006). "Methods for quantitative usability requirements: a case study on the development of the user interface of a mobile phone. ." Personal Ubiquitous Comput: 345-355.
42. Jokela, T., A. Seffah, et al. (2005). Guiding Designers to the World of Usability: Determining Usability Requirements Through Teamwork. Netherlands.
43. Juristo, N., A. M. Moreno, et al. (2007). "Guidelines for Eliciting Usability Functionalities". IEEE Transaction on Software Engineering: 744-758.
44. Kuzmina, N. V. (1970). " Metodías investigativas de la actividad pedagógica".
45. Leffingwell, D. and D. Widrig (2003). Managing Software Requirements: A Use Case Approach. Boston.
46. López, R. E. and J. P. Deslauriers (2011). "La entrevista cualitativa como técnica para la investigación en Trabajo Social".
47. Lorés, J., M. Sendín, et al. (2001). Evaluación. España, Universidad de Lleida.
48. Lowe, D. and J. Ekluind (2002). "Client Needs and the Design Process in Web Projects".
49. Mascheroni, M., C. Greiner, et al. (2012). "Calidad de software e Ingeniería de Usabilidad". XIV Workshop de Investigadores en Ciencias de la Computación.
50. Mendoza, L. E. (2012). "Sistema de Información II Teoría. Determinación de requerimientos entrevistas, cuestionarios, observaciones, JAD, prototipos, case, groupware".
51. Merlino, H. D. (2014). Inclusión de servicios en aplicaciones basados en patrones de usabilidad caso undo/redo.
52. Ministerio de economía y planificación (2012). Resolución No. 444/2012.
53. Moreno, A. M. and M. Sánchez-Segura (2007). "Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico".
54. Moreno, J. C. and M. M. Marciszack (2013). " La Usabilidad Desde La Perspectiva De La Validación de Requerimientos No Funcionales Para Aplicaciones Web".
55. Nielsen, J. (1993). "Usability Engineering." Academic Press Professional: 115-206.
56. Ormeño, Y. I., J. I. Panach, et al. (2012). "Elicitación de Requisitos de Usabilidad: Un estudio Sistemático". XIII Congreso Internacional de Interacción 2012.
57. Panach Navarrete, J. I. (2010). Incorporación de Mecanismos de Usabilidad en un Entorno de Producción de Software Dirigido por Modelos. Valencia, España, Universidad Politécnica de Valencia.
58. Pérez Olmo, Y. (2009). Técnicas y herramientas de la ingeniería de requisitos adecuadas para simuladores virtuales. La Habana, Universidad de las Ciencias Informáticas.

REFERENCIAS BIBLIOGRÁFICAS

59. Pérez Teruel, K. (2007). Modelo de Referencia para la Ingeniería de Requisitos en Proyectos de Bioinformática. La Habana, Universidad de las Ciencias Informáticas.
60. Perurena Cancio, L. and M. Moráguez Bergues (2013). "Usabilidad de los sitios Web, los métodos y las técnicas para la evaluación".
61. Pressman, R. S. (2002). Ingeniería de Software. Un enfoque práctico.
62. Pressman, R. S. and M. G. Hill (2009). Ingeniería del Software. Un enfoque práctico.
63. Pressman, R. S. and L. Joyanes Aguilar (1997). "Ingeniería del software: un enfoque práctico". 122.
64. Ramli, R. and A. Jaafar (2008). "A cheap possible solution for usability evaluation, Information Technology". International Symposium on Informatic Technology **3**: 1-5.
65. Ramos Blanco, K. (2013). Proceso Base de Ingeniería de Requisitos para las pequeñas y medianas empresas de desarrollo de software La Habana, Cuba, Universidad de las Ciencias Informáticas.
66. Robertson, S. and J. Robertson (2006). "Mastering the Requirements Process".
67. Rojo, S. (2012). "Requerimientos No Funcionales para Aplicaciones Web".
68. Rojo, S. and A. Oliveros (2012). "Requerimientos No Funcionales Para Aplicaciones Web". JAIIO.
69. Ruiz, J. (1996). Metodología de la investigación cualitativa. España.
70. Sampieri, R. H. and C. F. Collado (2008). Metodología de la investigación. Colombia.
71. Seffah, A. and E. Metzker (2004). "The obstacles and myths of usability and software engineering". Communications of the ACM: 71-76.
72. Shackel, B. (1991). Usability—context, framework, design and evaluation. Human Factors for Informatics Usability. . Cambridge.
73. Sindhgatta, R. and T. Srinivas (2005). "Functional and Non-Functional Requirements Specification for Enterprise Applications".
74. Soares, M. S. and J. L. Vrancken (2008). "Model-driven User Requirements Specification using SysML". Journal of Software: 57-68.
75. Software Engineering Institute (2010). CMMI for Development.
76. Sommerville, I. (2004). Software Engineering.
77. Sommerville, I. (2005). Ingeniería del Software. México DF.
78. Valles, M. (2003). "Técnicas Cualitativas de Investigación Social. Reflexión metodológica y práctica profesional".
79. Wiegers, K. E. (2003). Software Requirements, Redmond:Microsoft Press.

BIBLIOGRAFÍA CONSULTADA

1. Aceves, L.C. (2007). El estado de la usabilidad y accesibilidad en el mundo y en México.
2. Barrera. Y; Ruiz. M. (2010). Propuesta del perfil por competencias para el rol de Diseñador de Interfaz de Usuario en la UCI.
3. Basto Cordero, L. J., Ribeiro Parente Filho, L. F., Costa dos Santos, R., Gassenferth, W., and Soares Machado, M. A. (2013). Ipod System's Usability: An Application of the Fuzzy Logic. *Global Journal of Computer Science and Technology*, 13(6).
4. Bertoa, M. F., Troya, J. M. and Vallecillo, A. (2006). Measuring the usability of software components. *Journal of Systems and Software*, 79(3), 427–439. doi:10.1016/j.jss.2005.06.026
5. Bevan, N. and Macleod, M. (1994). Usability measurement in context. *Behaviour & Information Technology*, 13(1-2), 132–145. doi:10.1080/01449299408914592
6. Beyan, N. (1992). The MUSIC methodology for usability measurement. Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems. ACM, 123–124.
7. Bhatnagar, S., Dubey, S. K. and Rana, A. (2012). Quantifying Website Usability using Fuzzy Approach. *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307*, 2(2), 424–428.
8. Chang, E. and Dillon, T. S. (2006). A usability-evaluation metric based on a soft-computing approach. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(2), 356–372. doi:10.1109/TSMCA.2005.851349
9. Chang, E. J., Dillon, T. S. and Cook, D. (1997). An Intelligent System Based Usability Evaluation Metric. In *Intelligent Information Systems, IASTED International Conference on. IEEE Computer Society* (pp. 218–226).
10. Costa, R., Augusta, M., Machado, S., Santos, D. J., Tavares, R. and Campos, C. De. (2007). Uma Aplicação da Matemática Nebulosa na Usabilidade de Pacotes Estatísticos. *RESI – Revista Eletrônica de Sistemas de Informação*, 10(1), 1–8.
11. Dasso, A. and Funes, A. (2012). Software Quality Metrics Aggregation. *13th Argentine Symposium on Software Engineering, ASSE 2012 ISSN: 1850-2792*, 312–323.
12. Dubey, S. K., Gulati, A. and Rana, A. (2012). Usability Evaluation of Software Systems using Fuzzy Multi- Criteria Approach. *IJCSI International Journal of Computer Science Issues ISSN (Online): 1694-0814*, 9(3), 404–409.
13. Dubey, S. K., Mittal, A. and Rana, A. (2012). Measurement of Object Oriented Software Usability using Fuzzy AHP. *International Journal of Computer Science and Telecommunications*, 3(5), 98–104.

14. Escalona, M.J; Koch, N. (2006). Metamodeling the Requirements of Web Systems. International Conference on Web Information System and Technologies (WEBIST 2006). Portugal, pp. 310-317. Consultado: 05/07/2013.
15. Furer, B., Ruggeri, F., Voci, S. M., Borges, C. A. and Slater, B. (2013). Avaliação da usabilidade de um sistema computadorizado de epidemiologia nutricional. Bras Epidemiol, 16(4), 966–975.
16. Góngora, A. (2011). Catálogo automatizado de métricas de calidad para evaluar los productos en las pruebas.
17. Herrera, A M; Nariño, O. (2010). Procedimiento para evaluar la usabilidad de sistemas de gestión sobre plataformas web sin intervención del usuario final.
18. Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. International Journal of Human-Computer Studies, 64(2), 79–102. doi:10.1016/j.ijhcs.2005.06.002
19. <http://www.desarrolloweb.com/articulos/usabilidad-y-accesibilidad.html>
20. http://www.ecured.cu/index.php/Calidad_de_Software
21. <http://www.sedic.es/autoformacion/accesibilidad/11-usabilidad-accesibilidad.html>
22. <http://www.usabilidadb2c.mx/webu/usabilidadb2c/>
23. <http://www.usolab.com/wl/>
24. http://www.webusability.com.mx/webu/clients_es.htm
25. Hub, M. and Zatloukal, M. (2009). Usability Evaluation of Selected Web Portals 2 Case study – Usability evaluation of selected Web portals. Proceedings of the 9th WSEAS International Conference on APPLIED INFORMATICS AND COMMUNICATIONS (AIC '09), 259–264.
26. Hub, M. and Zatloukal, M. (2010). Model of Usability Evaluation of Web Portals Based on the Fuzzy Logic. WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS, 7(4), 522–531.
27. IEEE. (2009). IEEE Recommended Practice for Software Requirements Specifications. Disponible en: <http://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf>.
28. ISO/IEC 14598-1 (1998). Information Technology – Evaluation of Software Products – General Guide.
29. Jin, B. S. and Ji, Y. G. (2009). Development of a Usability Evaluation Framework with Quality Function Deployment: From Customer Sensibility to Product Design. Human Factors and Ergonomics in Manufacturing, 19(2), 177–194. doi:10.1002/hfm
30. Lamichhane, R. and Meesad, P. (2011). A Usability Evaluation for Government Websites of Nepal Using Fuzzy AHP. The 7th International Conference on Computing and Information Technology IC2IT2011, 99–104.

31. Li, Y. and Man, Z. (2008). A Fuzzy Comprehensive Quality Evaluation for the Digitizing Software of Ethnic Antiquarian Resources. 2008 International Conference on Computer Science and Software Engineering, 1271–1274. doi:10.1109/CSSE.2008.304
32. Liang, L., Deng, X. and Wang, Y. (2009). Usability Measurement Using a Fuzzy Simulation Approach. 2009 International Conference on Computer Modeling and Simulation, 88–92. doi:10.1109/ICCMS.2009.76
33. Madhavan, R. and Alagarsamy, K. (2014). USABILITY EVALUATION OF OBJECT ORIENTED APPLICATIONS USING LEARNING CURVES. International Journal of Engineering and Technical Research (IJETR), 2(2 ISSN: 2321-0869), 35–39.
34. Marzal, M.A; Calzada, J; Vianello, M. (2008). Criterios para la evaluación de la usabilidad de los recursos educativos virtuales: un análisis desde la alfabetización en información. Madrid, España. Vol. 13. No. 4.
35. Marzán, A; Villar, J C. (2010). Procedimiento para pruebas de software con herramientas automatizadas en el Departamento de Pruebas de Software.
36. Mich, L., Franch, M. and Gaio, L. (2003). Evaluating and Designing Web Site Quality. MultiMedia, IEEE, 10(1), 34–43.
37. Montazer, Gh. A. and Saremi, H. Q. (2008). An Application of Type-2 Fuzzy Notions in Website Structures Selection: Utilizing Extended TOPSIS Method. WSEAS TRANSACTIONS on COMPUTERS, 7(1), 8–15.
38. Nagpal, R., Mehrotra, D., Sharma, A. and Bhatia, P. (2013). ANFIS Method for Usability Assessment of Website of an Educational Institute. World Applied Sciences Journal, 23(11), 1489–1498. doi:10.5829/idosi.wasj.2013.23.11.790
39. Nielsen, J. (1994). Usability Engineering (Interactive Technologies).
40. Oztekin, A., Kong, Z. J. and Uysal, O. (2010). UseLearn: A novel checklist and usability evaluation method for eLearning systems by criticality metric analysis. International Journal of Industrial Ergonomics, 40(4), 455–469. doi:10.1016/j.ergon.2010.04.001
41. Pozo, D; Figueroa, L. (2010). Procedimiento general de evaluación de Usabilidad.
42. Preece, J. (2001). Sociability and usability in online communities: determining and measuring success. BEHAVIOUR & INFORMATION TECHNOLOGY, 2001, 20(5), 347–356. doi:10.1080/0144929011008468
43. Santos, D. J., Tavares Carneiro de Campos, R., Parente Filho, L. F., Costa dos Santos, R. and Soares Machado, M. A. (2007). Usabilidade de sistemas: uma aplicação da lógica fuzzy na avaliação do PHSTAT.
44. Schenkman, B. N. and Jönsson, F. U. (2000). Aesthetics and preferences of web pages. Behaviour & Information Technology, 19(5), 367–377. doi:10.1080/014492900750000063

45. Shawn, H. (2008). Simplemente pregunta: Integración de la accesibilidad en el diseño. Disponible en: www.uiAccess.com/JustAsk/es/ut.html
46. Sicilia, M. and García, E. (2003). Modelling Interacting Web Usability Criteria through Fuzzy Measures. *Web Engineering*. Springer Berlin Heidelberg, 182–185.
47. Sicilia, M.-ángel, Barriocanal, E. G. and Calvo, T. (2003). An inquiry-based method for Choquet integral-based aggregation of interface usability parameters. *Kybernetika*, 39(5), 601–614.
48. Singh, A. and Dubey, S. K. (2013). Evaluation of Usability Using Soft Computing Technique. *International Journal of Scientific & Engineering Research*, 4(12), 162–166.
49. Sitar-Taut, D.-A., Stanca, L., Buchmann, R. and Lacurezeanu, R. (2009). A Case Study on Usability Metrics Applied in Romanian E-Commerce Environment. *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS*, 6(10), 1697–1706.
50. Siu Keung, W., Thi Nguyen, T., Chang, E. and Jayaratna, N. (2003). Usability Metrics for E-learning. In *On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops*. Springer Berlin Heidelberg (pp. 235–252).
51. Software Engineering Institute. (2005). *Process Maturity Profile*.
52. Sommerville, I. (2007). *Software engineering*. Octava edición ed., Harlow, Ed.: Addison Wesley.
53. Tamir, D., Komogortsev, O. V. and Mueller, C. J. (2008). An Effort and Time Based Measure of Usability. *Proceedings of the 6th International Workshop on Software Quality*. ACM, 47–52.
54. Winter, S., Wagner, S. and Deissenboeck, F. (2008). A Comprehensive Model of Usability. *Engineering Interactive Systems*. Springer Berlin Heidelberg, 106–122.

ANEXOS

Anexo 1: Evaluación de proyectos según CHAOS Reports-Standish (STANDISH GROUP, 2013)

Evaluación	1994	1996	1998	2000	2002	2004	2006	2008	2010	2012
Exitosos	16%	27%	26%	28%	34%	29%	35%	32%	37%	39%
No satisfactorio	53%	33%	46%	49%	51%	18%	19%	24%	21%	18%
Cancelados	31%	40%	28%	23%	15%	53%	46%	44%	42%	43%

Anexo 2: Encuesta presentada para definir la situación problemática

Años de experiencia: _____

Cargo o Responsabilidad: _____

Grado científico: _____

Entidad: _____

Nota: El objetivo de esta encuesta es solamente investigativa. El responsable de la misma se compromete a mantener total privacidad de la información recopilada.

1. ¿En su proyecto se gestionan y desarrollan correctamente los requisitos de usabilidad?

Sí _____ No _____

2. Seleccione con una X los roles que deben participar en el desarrollo de los requisitos de usabilidad.

____ Administrador de la Calidad

____ Analista de Software

____ Cliente

____ Arquitecto de Información

____ Arquitecto de Software

____ Jefe de Proyecto

____ Otros (Cuáles)

3. Seleccione con una X los roles que deben participar en el desarrollo de la arquitectura de usabilidad.

____ Administrador de la Calidad

____ Analista de Software

____ Cliente

____ Arquitecto de Información

____ Arquitecto de Software

____ Jefe de Proyecto

____ Otros (Cuáles)

4. De acuerdo a su rol en el proyecto, qué responsabilidades usted considera que no deben faltar en la Ingeniería de Usabilidad.

Argumente _____

5. ¿Qué fases considera que deben aplicarse para el desarrollo de los requisitos de usabilidad?

Marque con una X.

____ Elicitación

____ Análisis

____ Especificación

____ Verificación

____ Gestión

6. ¿Qué técnica(s) utiliza para desarrollar los requisitos de usabilidad? ¿En qué fase(s) lo(s) emplea?

___ Entrevistas

___ Talleres

___ Escenarios

___ Pseudocódigo

___ Historias de usuario

___ Tablas y árboles de decisión

___ Modelo Entidad Relación

___ Tormenta de ideas

___ Lista de chequeo

___ Prototipos

___ Casos de prueba

___ Construcción de prototipos

___ Casos de uso

___ Cuestionarios

___ Otra(s) ¿Cuál? (es) _____

7. ¿Utiliza algún patrón de usabilidad para mejorar la usabilidad en el desarrollo de los productos de software? En caso afirmativo diga cuáles.

Sí _____ No _____

Anexo 3: Encuesta presentada al grupo de expertos para validar la solución propuesta

Años de experiencia: _____

Cargo o responsabilidad: _____

Grado científico: _____

Entidad: _____

Nota: El objetivo de esta encuesta es solamente investigativa. El responsable de la misma se compromete a mantener total privacidad de la información recopilada.

1. Valore la metodología web y la herramienta NDT-Suite propuesta para el correcto desarrollo de los requisitos de usabilidad.
 Muy adecuada
 Adecuada
 Algo adecuada
 Poco adecuada
 Inadecuada
2. ¿Es posible cumplir con las actividades y artefactos propuestos?
 En todos los casos
 En la mayoría de los casos
 En algunos casos
 En la minoría de los casos
 En ningún caso
3. Valore la propuesta de los patrones relacionados con el desarrollo de los requisitos de usabilidad.
 Muy adecuada
 Adecuada
 Algo adecuada
 Poco adecuada
 Inadecuada
4. ¿Considera usted que los requisitos de usabilidad definidos son:
 Muy importante
 Importante
 Algo importante
 Poco importante
 Nada importante

5. Valorar si los roles y responsabilidades son suficientes y necesarios para el correcto desarrollo de los requisitos y patrones de usabilidad.

- Muy adecuado
- Adecuado
- Algo adecuado
- Poco adecuado
- Inadecuado

Anexo 4: Escala de Likert para valoración de expertos

Criterios de evaluación			Puntuación
Muy adecuada	Muy importante	En todos los casos	5
Adecuada	Importante	En la mayoría de los casos	4
Algo adecuada	Algo importante	En algunos casos	3
Poco adecuada	Poco importante	En la minoría de los casos	2
Inadecuada	Nada importante	En ningún caso	1

Anexo 5: Encuesta de satisfacción de usuario

Nota: El objetivo de esta encuesta es con la finalidad de conocer la satisfacción de los especialistas acerca de la solución propuesta. Sus criterios serán de vital importancia de ahí que se le pide ser lo más claro y sincero posible en sus respuestas. Se le garantiza total confidencialidad de la información recopilada y anonimato.

1. ¿Considera que la incorporación de la Ingeniería de Usabilidad en el proceso de desarrollo del software es complejo y difícil de entender?

Sí No No sé

2. ¿Si Ud. fuera a realizar otro proyecto utilizaría los requisitos de usabilidad propuestos así como los patrones de usabilidad?

Sí No No sé

3. ¿Le satisface sus expectativas la solución propuesta?

- Me gusta mucho
- Me gusta más de lo que me disgusta
- Me es indiferente
- Me disgusta más de lo que me gusta
- No me gusta
- No puedo decir

4. ¿Considera útil la solución propuesta para introducir la gestión de los requisitos y patrones de usabilidad en el proceso de desarrollo del software?
5. ¿Tiene alguna recomendación acerca de la solución propuesta?

Anexo 6: Grupo Focal. Guía de desarrollo

Número de participantes: 7

Fecha: 3 de marzo de 2015

Lugar: Laboratorio 1 de la Infraestructura Productiva

Hora: 10:00 am

Apertura

Describir lo que constituye un grupo focal. Explicar el objetivo de la reunión.

Objetivo: Validar la propuesta de solución de la investigación emitiendo criterios sobre los aspectos tratados.

Guía de preguntas:

1. ¿Creen factible que la solución propuesta sea adaptable en los proyectos productivos?
2. ¿Consideran necesario la gestión de los requisitos de usabilidad en el proceso de desarrollo del software?
3. ¿Consideran que los requisitos de usabilidad son flexibles debido a su carácter genérico en la definición?
4. ¿Consideran que la incorporación del diseñador y el arquitecto de información en la solución propuesta, contribuya a disminuir la cantidad de NC de usabilidad de los productos finales?
5. ¿Incluirían o modificarían alguna actividad descrita en la definición de los requisitos de usabilidad?
6. ¿Incorporarían o modificarían técnicas, herramientas y metodología de la propuesta?
7. ¿Los patrones de usabilidad incluidos en la solución propuesta los modificaría?

Anexo 7: Entrevista a profundidad

Nota: El objetivo de esta entrevista es solamente investigativa. El responsable de la misma se compromete a mantener total privacidad de la información recopilada.

1. ¿Considera que la propuesta cubre las necesidades presentes en los proyectos productivos?
2. ¿Considera factible que la propuesta pueda ser extendida y utilizada en los proyectos productivos?
3. ¿Considera que el procedimiento propuesto concibe la relación entre las actividades y los artefactos de entrada y salida, los roles y responsabilidades, técnicas, metodologías y herramienta

necesaria a cumplimentar para definir los requisitos y patrones de usabilidad en los proyectos productivos?

4. ¿Considera apropiadas y suficientes las actividades descritas en la propuesta de solución?

5. ¿Incluiría o modificaría algún rol en la incorporación de la Ingeniería de Usabilidad del proceso de desarrollo del software? Si la respuesta es Sí argumente cuáles.

Sí ___ No ___

6. ¿Incluiría o modificaría alguna responsabilidad de los involucrados en la incorporación de la Ingeniería de Usabilidad en el proceso de desarrollo del software? Si la respuesta es Sí argumente cuáles.

Sí ___ No ___

7. ¿Recomendaría esta solución propuesta a otras personas interesadas en el tema?

Sí ___ No ___ No sé___

8. ¿Qué consideras que puede mejorar en la solución propuesta?

Todo___ Una parte___ Nada___

Cuál: _____.

9. ¿Considera que la solución propuesta concibe la relación entre los atributos, propiedades y patrones de usabilidad, según la norma 25010: 2011?