

ENFOQUE METODOLÓGICO PARA EL DISEÑO DE INTERFACES DURANTE EL CICLO DE VIDA DE DESARROLLO DE SOFTWARE

METHODOLOGICAL APPROACH FOR INTERFACE DESIGN DURING SOFTWARE DEVELOPMENT LIFE-CYCLE



AUTOR

JAVIER M. MARTÍNEZ GÓMEZ

Ph.D. Diseño industrial

M.Sc. Informática

*Universidad Industrial de Santander

Profesor Asistente

Escuela de Diseño industrial

javimar@uis.edu.co

COLOMBIA

AUTOR

MIGUEL E. HIGUERA MARÍN

M.Sc. Desarrollo sostenible

Esp. En Producción

*Universidad Industrial de Santander

Profesor Asistente

Escuela de Diseño Industrial

ehiguera@uis.edu.co

COLOMBIA

AUTOR

ESPERANZA AGUILAR DÍAZ

M.Sc. en Física

M.Sc. en Pedagogía

*Universidad Industrial de Santander

Profesora Titular

Escuela de Física

eaguilar@uis.edu.co

COLOMBIA

*INSTITUCIÓN

Universidad Industrial De Santander

UIS

Universidad Publica

Cra.27 Calle 9 Ciudadela

Universitaria, Bucaramanga

webadmin.@uis.edu.co

COLOMBIA

INFORMACIÓN DE LA INVESTIGACIÓN O DEL PROYECTO: Este artículo describe algunos de los resultados de investigación del proyecto "Enfoque metodológico para el diseño de interfaces" desarrollado por los grupos de investigación GEMA e INTERFAZ de la Universidad industrial de Santander.

RECEPCIÓN: Agosto 1 de 2013

ACEPTACIÓN: Septiembre 18 de 2013

TEMÁTICA: Ingeniería de software, gestión de la calidad de proyectos

TIPO DE ARTÍCULO: Artículo de Investigación Científica e Innovación

RESUMEN ANALÍTICO

La ingeniería de software busca definir procesos, controlarlos y seguirlos con el fin de construir productos de calidad que garanticen al usuario su satisfacción, generalmente medida en términos de funcionalidad del producto, pero pocas veces evaluada en términos de usabilidad del mismo. Este artículo presenta un modelo para el ciclo de vida de desarrollo de software, basado en la metodología Human-Centered Design (HCD) en el cual son integrados algunos métodos utilizados en el diseño de producto (diseño industrial). Este modelo incluye métodos de usabilidad (eficiencia, eficacia y satisfacción) y técnicas para el diseño de interfaces gráficas de usuario (apreciación y estética). La investigación utilizó una metodología de análisis cualitativo a través del análisis de la experiencia de una PYME de desarrollo de software. Por último, se utilizó el modelo de referencia Capability and Maturity Model Integrated (CMMI) para la validación del modelo propuesto.

PALABRAS CLAVES: Usabilidad, Diseño Centrado en el Usuario, Diseño como Proceso, Ciclo de Vida del Producto.

ANALYTICAL SUMMARY

Software engineering looks for defining, controlling and following processes, it aiming to construct high quality products in order to guarantee user satisfaction. Usually, it is measured in functionality terms, but few times measured from the perspective of usability. This paper presents a life cycle model of software development, based on the Human Centered Design (HCD) methodology, which integrates some methods used in product design (industrial design). The model includes usability methods (efficiency, efficacy and satisfaction) and techniques for graphic user interfaces (aesthetic and visual perception). The research uses an analytic qualitative methodology approach by the experience of a SME (Small and Medium Enterprises) of software development. Finally, Capability and Maturity Model Integrated (CMMI) was used as reference for validating the model.

KEYWORDS: Usability, Human-Centered Design, design as process, product life cycle.

INTRODUCCIÓN.

Los computadores y el software son herramientas cuyo propósito es ayudar a las personas a realizar una tarea orientada a cumplir con determinados objetivos (trabajo, comunicación, diversión, etc.). Toda herramienta para poder ser empleada necesita comunicar al usuario sus posibilidades de uso, informar acerca de las acciones que son posibles, el estado actual del sistema y los cambios producidos, y esto lo hacen mediante su interfaz [1]. La medida de la eficiencia, eficacia y satisfacción con la que la herramienta ayuda al usuario a conseguir sus objetivos específicos, en un determinado contexto de uso se denomina usabilidad; el concepto de usabilidad está ampliamente definido en la norma ISO 9241-11 [2]. En Ingeniería de Software (IS) la disciplina que involucra el estudio, la planeación y diseño de sistemas de interacción entre seres humanos y computadoras se conoce como Interacción Hombre-Computadora (HCI).

Debido a la demanda de productos con interfaces claras y comprensibles, que permitan acceder sin frustración a todos los beneficios ofrecidos por los sistemas [3] y el creciente uso de productos informáticos tales como: páginas web, videojuegos, aplicaciones en dispositivos móviles, etc., ha despertado un interés en las empresas productoras de software por construir productos con una mejor usabilidad.

Numerosos han sido los inconvenientes que se generan con los productos por falta de usabilidad: sobre costos por correcciones constantes, aumento en los tiempos de desarrollo, insatisfacción de los clientes y rechazo del producto, como lo muestra la clasificación en problemas de usabilidad descrita en [4].

Los procesos de Ingeniería de Software buscan productos de calidad que satisfagan a los usuarios, pero tradicionalmente se han concentrado casi exclusivamente

en aspectos de la funcionalidad del producto, olvidando que un producto con aceptación en el mercado igualmente debe gozar de otras características como: facilidad de uso, facilidad de aprendizaje, utilidad y apreciación estética [5].

El trabajo de proyectar interfaces con una mejor usabilidad en el mundo de los artefactos y bienes de consumo como autos, herramientas, mobiliario, etc., ha sido tradicionalmente una labor de los diseñadores industriales. Sin embargo, en el mundo del software (producto intangible) solo hasta la popularización de Internet se inicia el reconocimiento a la contribución que profesionales en diseño pueden ofrecer, básicamente, por dos ideas erróneas acerca del diseño: 1) Que existen dos tipos de diseño: el diseño de ingeniería y el diseño como un asunto de estética 2) que el diseño de interfaces es sólo un asunto de gráficos, decoración y dibujos independientes de la funcionalidad del producto [6] 3) Que las interfaces se reducen a los componentes con los que un usuario manipula un producto (capa de presentación). Es por esta concepción errónea del diseño que se desaprovecha el potencial de las contribuciones que puede ofrecer el trabajo de un diseñador durante todo el ciclo de vida del desarrollo de software.

Cualquiera sea el proceso o metodología usada para el desarrollo de software, siempre se debe aplicar un modelo de ciclo de vida. Ahora bien, ¿Cómo es posible integrar el trabajo de diseñadores e ingenieros en el diseño y desarrollo de productos informáticos con mejores interfaces durante todo el ciclo de vida?, este artículo presenta un modelo de ciclo de vida que integra el enfoque de métodos tradicionalmente utilizados en el diseño de producto (diseño industrial) al desarrollo de software.

En la actualidad se han propuesto diversas agrupaciones de técnicas de diseño que son aplicadas en productos informáticos con el objetivo de mejorar la experiencia del usuario (UX) con el producto en diferentes contextos, p.ej. la guía para el diseño de interfaces propuesta por Monique W. M. en [7], una propuesta para el diseño de interfaces de software médico en oncología; o el método para el diseño de interfaces en productos multimedia propuesto por Sutcliffe en [8], sólo por nombrar algunos. Todos estos modelos demuestran múltiples beneficios, donde principalmente destacan como resultado el incremento en la satisfacción y eficiencia en el desarrollo de las actividades de los usuarios. En el campo del HCI algunas herramientas y métodos para guiar el proceso de desarrollo han sido descritas por Shneiderman en [9]. Sin embargo, todos aún consideran el rol del diseño como una parte del proceso y no como parte inherente al proceso mismo.

No obstante de los grandes esfuerzos y avances significativos que en la industria informática local se están haciendo para el desarrollo de productos con una mejor usabilidad como las técnicas CIAT-GUI que nos presenta Giraldo en [10], la información proveniente de las prácticas de diseño de interfaces no se administra ni se integra con otras áreas de proceso, y se gestiona de manera independiente de los demás procesos de desarrollo. Esto genera problemas tales como: difícil acceso a la información y su reúso en nuevos proyectos, trazabilidad de los requisitos de usabilidad, identificación de las pruebas necesarias para saber si un producto está cumpliendo con los requerimientos de facilidad de uso, conocer con anticipación el impacto que el cambio en un requisito de usabilidad tiene sobre otras áreas del proceso de desarrollo y en otros componentes del producto.

Este artículo presenta una experiencia de la descripción y proceso de implementación de un modelo para el desarrollo de interfaces donde ingenieros, analistas, diseñadores, arquitectos de software y las personas que utilizan los productos se integran por medio del uso de métodos de diseño a lo largo de todo el ciclo de vida del producto. El modelo presentado en esta contribución describe las principales prácticas en diseño (algunas ya comunes en contextos informáticos) y propone un sistema para gestionar la información proveniente de estas actividades durante todo el ciclo de vida e integrarla con otras áreas de proceso, contribuyendo de esta manera a una gestión eficiente de la información de diseño y a obtener una mejor aceptación y percepción del producto por parte de los usuarios finales. En especial muestra como fue aplicado en un contexto empresarial a nivel local.

La investigación se desarrolló en tres fases: En la primera fase se realizó un estudio exploratorio de los modelos de IS con el objetivo de identificar los métodos de diseño de producto que pueden ser usados en cada una de las fases del ciclo de vida del desarrollo de software; en la segunda parte, se propuso un enfoque metodológico para el desarrollo de software orientado al diseño como proceso en todo el ciclo de vida y no como una actividad; en la tercera etapa se propuso un ciclo de vida para el desarrollo de software, el cual fue implementado como una propuesta de mejoramiento continuo mediante el desarrollo de tres proyectos de desarrollo de software, en la empresa FCV.Soft. Unidad de Negocio de la Fundación Cardiovascular de Colombia. Por último, el modelo se validó y demostró su innovación¹ usando como referencia el modelo CMMI que permitió tener un diagnóstico preciso de la madurez de los

¹La innovación entendida como la aplicación de mejores soluciones que satisfacen necesidades desarticuladas o existentes en un mercado concreto.

procesos planteados. La duración total del estudio fue de 32 meses.

Uno de los aspectos que diferencia este modelo de otros que buscan el mismo fin, es que al ser evaluado mediante las prácticas de CMMI significa que la información proveniente de las actividades del proceso de diseño fue sometida al proceso de gestión de la configuración y administración de cambios, es decir, permitió su integración y trazabilidad durante todo el ciclo. En otras palabras, permitió asociar cada requerimiento de usabilidad a características y componentes específicos del producto y estos a su vez enlazados con pruebas y test, lo que consiente visualizar el impacto que una modificación en un requerimiento de interfaz tenía sobre otros requerimientos y sobre otros procesos.

1. MÉTODOS DE DISEÑO DE PRODUCTO EN INGENIERÍA DE SOFTWARE.

En la industria del software al igual que en la producción de otros tipos de productos (tangibles o intangibles), diseño hace referencia a todos los procesos que hacen posible el desarrollo del producto en las etapas de concepción, definición, construcción y puesta en funcionamiento. Es decir, diseño de producto va más allá de la tradicional etapa mencionada en la mayoría de los modelos de ingeniería de software, tales como: modelo en cascada, modelo en espiral, desarrollo por etapas, etc. [11], donde diseño hace referencia sólo a la fase en la cual el sistema se define y detalla antes de la programación.

Por consiguiente, diseño es una actividad que debe ser desarrollada a lo largo de todo el ciclo de vida y los métodos de diseño deben ser orientados hacia los usuarios finales y los objetivos que ellos quieren conseguir de las herramientas, además de incorporar análisis de tareas y contexto de uso, es decir, factores de usabilidad que además de garantizar la funcionalidad del producto, asocian aspectos humanos al desarrollo de la herramienta como lo afirma Poelman en [12].

Los métodos de diseño permiten crear productos de calidad, innovadores y con gran aceptación en el mercado; y los beneficios de incorporar diseñadores son aplicables a lo largo de todo el ciclo de vida del producto, desde las etapas de ideación, concepción, definición, producción, comercialización y distribución, uso y mantenimiento, hasta su disposición final, este enfoque es conocido como Diseño como proceso [13].

Para esta investigación se tomaron como base los

paradigmas de los modelos descriptivos y los modelos prescriptivos. Los modelos descriptivos para establecer las secuencias de actividades que hacen énfasis en generar un concepto de solución en etapas tempranas del proceso enfocado en la solución de un problema o necesidad. Después, esto es analizado, evaluado, refinado y desarrollado, llevando el proceso en la dirección correcta a través del estudio de necesidades, al análisis y planteamiento del problema, el desarrollo de un diseño conceptual, la comunicación del diseño y el desarrollo de detalles. Por otra parte, los modelos prescriptivos buscan adoptar mejores formas de trabajar mediante un proceso sistemático a seguir, proporcionado a través de una metodología de diseño particular, para esta investigación se eligió la metodología *Human Centered Design* (HCD) por ser un enfoque que toma como base el modelo prescriptivo y al igual que en IS su énfasis se basa en la especificación de requerimientos derivados del problema de diseño, donde se generan conceptos alternativos, desarrollando soluciones secundarias y haciendo una elección objetiva del mejor de los diseños alternativos, pero poniendo al usuario como protagonista de todo el proceso de validación [14].

El HCD es una perspectiva más amplia del denominado *User Centered Design* (UCD), la diferencia entre los dos enfoques radica en que HCD parte de la premisa que no solo se deben tratar los seres humanos como simples usuarios de un producto, sino que las personas hacen parte de un sistema complejo mas amplio que involucra otros aspectos tales como: puestos de trabajo, factores humanos, relaciones laborales, medio ambiente, etc. En Ingeniería de Software esto cobra gran relevancia ya que la UX no inicia cuando las personas están sentadas frente a la computadora. La UX con un producto informático inicia desde los procesos de compra e instalación y se ven influenciados por diferentes aspectos como el packaging, las características del ambiente donde será usado el software (iluminación, ruido, temperatura, etc.) y hasta por las características psicológicas de las personas que lo usarán.

De conformidad con las etapas del ciclo de vida en el desarrollo de software: análisis de requerimientos, definición del producto, construcción, evaluación e implantación; se identificaron los principales métodos de diseño propuestos en el enfoque metodológico de HCD [15] y la recopilación de estrategias para el diseño de producto propuesta por Niegel Cross en [16]; para después, identificar el rol del diseño en cada etapa.

A continuación se realiza una descripción de lo que para el modelo se entiende por diseño en cada fase, los métodos utilizados y como son aplicados.

1.1 DISEÑO DURANTE EL ANÁLISIS DE REQUERIMIENTOS.

Generalmente, el punto de partida en la concepción de un nuevo producto es un problema definido de manera amplia; según M. Jones [17] uno de los principales problemas al inicio de cada proyecto de software es la mala traducción de las necesidades de los usuarios en requisitos del producto.

En consecuencia el primer paso es tratar de clarificar los objetivos y hacer una especificación de requerimientos con los usuarios construyendo un mapa mental de lo que ellos esperan y desean del producto.

Los principales métodos de diseño identificados en etapas tempranas de desarrollo son los métodos de indagación, dentro de los cuales existen diversas formas de aproximación con el usuario: aproximación contextual (indagación en el contexto y estudios etnográficos), aproximación por grupos (grupos orientados, grupos de debate, lluvias de ideas), aproximación individual (encuestas, cuestionarios) y remota (encuesta remota, sesiones guiadas, registro por el usuario).

Para la generación de ideas se usan técnicas como: secuencia de escenarios, creación de escenarios, cuadros de organización de tareas, análisis de tareas y matrices de funcionalidad.

1.2 DISEÑO DURANTE LA DEFINICIÓN DEL PRODUCTO.

La actividad esencial del diseño durante la definición del producto, es la producción de una especificación final y detallada del producto, comprensible para producción. En la industria manufacturera esta especificación se hace a través de dibujos y planos técnicos, en IS estos dibujos podrían abarcar desde descripciones generales que dan una visión amplia del producto, hasta las más específicas que proporcionan instrucciones precisas de cómo va a funcionar el producto y qué consideraciones se deben tener en cuenta para su elaboración (*storyboarding*) [18]. Otro de los propósitos de los dibujos es la inspección y verificación de las propuestas antes de decidir sobre una versión final para implementación. La principal razón para mantener separados los procesos de definición de producto, de los de producción radica en que las propuestas puedan ser verificadas y analizadas antes de ser enviadas a fabricación, esto mismo vale para los procesos de IS.

Mientras que en las etapas de análisis se trata de entender el problema en la etapa de definición se trata de encontrar la solución. Problema-solución se desarrollan

simultáneamente, esta exploración se realiza mediante bosquejo de ideas tentativas, esto es necesario debido a que normalmente el problema no está bien definido y está mal estructurado presentando las siguientes características:

- No existe una formulación definitiva del problema.
- Cualquier formulación del problema puede contener inconsistencias. Las formulaciones del problema dependen de la solución.
- La propuesta de soluciones es una forma de entender el problema.
- No existe una solución definitiva del problema.

Los métodos utilizados en esta etapa son técnicas como: Árbol de objetivos, análisis de funciones, especificación de rendimiento, despliegue de la función de la calidad (QFD), diagramas morfológicos, ponderación de objetivos e ingeniería del valor.

1.3 DISEÑO EN LA CONSTRUCCIÓN.

Una vez definido el producto este pasa a su fase de producción, en IS esta comprende todas las actividades de desarrollo, dentro de las cuales una es el diseño de la Interfaz Gráfica de Usuario (GUI), su meta es destacar en la interfaz el diseño del comportamiento del producto (interacción) para que el usuario comprenda el esquema lógico de la aplicación (arquitectura de información). Los principales paradigmas en este sentido son: WISIWIG (What You See Is What You Get), manipulación directa e interfaz basada en iconos. También existen modelos para la generación automática de los elementos visuales de la interfaz que focalizan la interacción del producto en la generación de controles y su distribución para la correcta visualización en diferentes plataformas: computadoras personales, celulares y dispositivos móviles, etc.

No obstante debemos recordar que el proceso de "Diseño visual" abarca muchos aspectos que se basan en la teoría de comunicación y percepción visual y son particulares para cada tipo de usuario teniendo en cuenta cuestiones como discapacidades físicas (accesibilidad), aspectos culturales (religión, tradiciones, lenguaje, etc.) y segmentación por grupos focales (sexo, edad, nivel socio-económico, etc.).

Los diseñadores elaboran la GUI con base en una serie de reglas y conceptos fundamentados en la semiótica, la percepción humana y la psicología cognitiva que hacen posible que las personas se mantengan orientadas durante la navegación, comprendan las acciones que son posibles y reconozcan el estado del sistema; para esto se conjugan conceptos sobre elementos de la imagen (el punto, la línea, la forma, el color, la luz, el

tiempo, el tamaño, el formato), técnicas de composición visual (disposición, énfasis, foco, alineación), técnicas de distribución de los elementos gráficos (sección áurea, ley de Fitts, serie de Fibonacci).

Además de los elementos de la GUI se deben relacionar también otros aspectos de la percepción que puedan hacer parte de la experiencia sensorial del usuario con el sistema tales como sonidos, olores, sensaciones táctiles, etc. No obstante, como anteriormente se dijo, la Experiencia de Usuario (UX) con el sistema inicia antes de que él este en frente de la computadora. La UX comprende también todo el saber previo: proceso de compra, instalación del software, capacitación, guías de uso, ayudas, etc. Por tanto el trabajo de los diseñadores en esta etapa incluye todo lo referente al diseño del material de ayuda y a otros elementos de comunicación que no están relacionados directamente con la interfaz; de manera general esto se conoce como Art work y comprende cuestiones como: manuales de usuario, sistemas de packaging, sistemas de ayuda (online y offline), material de promoción y publicidad (brochures), etc.

1.4 DISEÑO EN LA EVALUACIÓN.

La evaluación de software comprende principalmente dos actividades que buscan garantizar la calidad del producto final: la verificación y la validación.

El propósito de la verificación es asegurar que el producto reúne las especificaciones de los requerimientos técnicos, mientras la validación tiene como propósito demostrar que el producto satisface su uso, previsto en condiciones de uso reales [19].

- En la etapa de verificación los métodos de diseño más comunes son los de inspección. Estos métodos tienen el objetivo de formalizar el registro y descubrimiento de los problemas de software. Las inspecciones también proporcionan medidas cuantitativas que pueden ser seguidas mediante métodos estadísticos para el control de procesos. Existen varios tipos de técnicas que se pueden catalogar como inspecciones: inspección de características, inspección de consistencia, inspección de estándares e inspecciones formales de usabilidad. Son estas últimas las que se llevan a la práctica con más frecuencia [20].

Otros métodos de inspección que utilizan una serie de listas de principios y características a ser evaluadas son las llamadas Evaluaciones Heurísticas, los Paseos Cognitivos y las listas de comprobación (Checklist), generalmente usadas en combinación con otros métodos de inspección de usabilidad [21].

- En la etapa de validación se destaca la pertinencia del trabajo de los diseñadores dentro de la evaluación de productos informáticos por ser aquí donde se valida el producto en condiciones reales de uso, con usuarios verdaderos, que desarrollan tareas concretas. Es decir, se evalúa la usabilidad de la herramienta. El método más popular para esto es el Test de Usabilidad, método que consiste en desarrollar experimentos para obtener información específica acerca de una herramienta, a partir de la observación sistemática de usuarios llevando a cabo tareas reales. En el Test de Usabilidad se usan protocolos que permiten obtener diferentes tipos de información: protocolo de pensamiento manifestado, protocolo de preguntas, protocolo de descubrimiento conjunto, realización de medidas, análisis de expresión facial y seguimiento de mirada (Eyetracking) [5].

Uno de los aspectos más importantes cuando se pretende validar un producto informático, es la definición de las métricas que se usarán para evaluar la usabilidad, ya que cada producto tendrá las suyas y no son una fórmula que se puede repetir de la misma forma para cada proyecto. Estas medidas de usabilidad se categorizan de conformidad con los objetivos de la herramienta y pueden ser cualitativas (percepción de la satisfacción), definidas mediante variables nominales u ordinales; o cuantitativas (eficacia y eficiencia), definidas mediante variables de intervalos o rangos. De la naturaleza de las variables dependerá la validez estadística de los datos obtenidos.

1.5 DISEÑO EN LA IMPLANTACIÓN.

Al tratar de llevar un producto al contexto real de uso es necesario conocer el ambiente tecnológico en el que se desplegará además de tener en cuenta las consideraciones del grupo de personas que lo van a utilizar para garantizar la satisfacción de las demandas de los usuarios; no sólo desde el punto de vista técnico, sino del sentido humano que debe tener la aplicación. El diseñador es el encargado de las estrategias de aceptación social del producto, además del análisis de los puestos de trabajo y otras variables que pueden alterar el rendimiento de la herramienta, como lo son: los factores ambientales (luz, ruido, temperatura) y los factores ergonómicos (diseño del puesto de trabajo, antropometría, visibilidad, etc.) [22].

Es claro entonces que los métodos formales con los que los diseñadores trabajan durante el entero ciclo de vida del producto se pueden adaptar a cualquier modelo de IS. Las principales ventajas de integrar métodos de diseño en los procesos de desarrollo de software radican en:

- Aumentar la calidad del producto, hacer más previsibles los cambios.
- Mejorar la especificación de requerimientos.
- Optimizar la comunicación de los diferentes grupos de trabajo.
- Mejorar el aspecto estético-formal de la interfaz y los sistemas de comunicación e interacción del software con el usuario.
- Reducir los tiempos de producción (se llega más rápido a las soluciones requeridas).
- Aumento de la satisfacción de los usuarios.
- Mejora de la usabilidad de la herramienta [12].

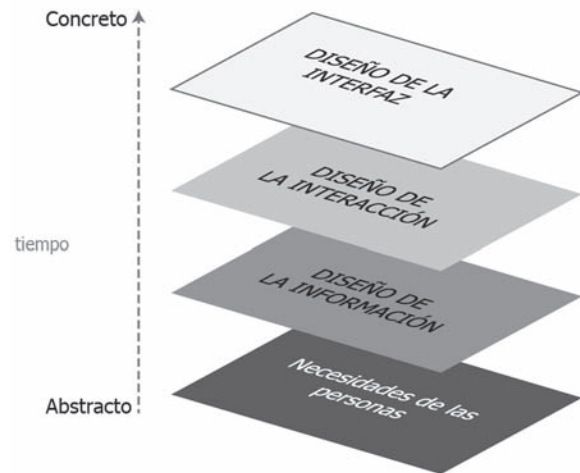
2. ENFOQUE METODOLÓGICO PARA EL DESARROLLO DE SOFTWARE.

Cada proyecto tiene objetivos, contenidos y soluciones creativas únicas; ver en el desarrollo de software el diseño como un proceso, es ver el proyecto como una suma de sus partes y las relaciones entre ellas, donde la clave es el diseño, donde el diseño está presente como soporte a todas las áreas de proceso de la organización. En este artículo se propone un enfoque metodológico enfocado hacia el diseño que comprende cuatro aspectos esenciales como se muestra en la Figura 1. La solución debe partir de las necesidades de la gente; después se piensa conceptualmente (diseño de la información) y en términos de comportamiento (diseño de la interacción); y finalmente (como consecuencia de un trabajo de diseño preliminar) se realiza la interfaz y los elementos gráficos.

2.1 DISEÑO DE LA INFORMACIÓN.

Diseño de la información significa clarificar cuales son los objetivos de la herramienta, establecer los requerimientos de información del producto, organizar y jerarquizar el contenido, definir el público que usará la herramienta, identificar el contexto de uso y establecer una arquitectura de información. Uno de los instrumentos que permite al final la observación de todos estos aspectos son los diagramas de flujo, en los cuales podemos apreciar los diferentes niveles de información, las categorías y la organización de la información que contiene la herramienta. Para esto podemos usar lenguajes de modelado ampliamente difundidos en IS como: UML (Unified Modeling Language) o IDEF (Integration DEFinition), solo por nombrar algunos. Soportados por mapas mentales, cuadros sinópticos, diagramas de secuencias, diagramas de procesos, etc.

FIGURA 1. Enfoque de desarrollo enfocado al diseño.



Para el modelo propuesto se usaron los diagramas de actividades de UML para especificar los procesos para convertirlos en una segunda iteración en casos de uso una vez estaban completamente definidos. Para la construcción de estos diagramas se usaron las técnicas descritas anteriormente en el numeral 1.1 con iteraciones constantes entre los analistas y los usuarios finales.

2.2 DISEÑO DE LA INTERACCIÓN CON EL USUARIO.

Diseñar la interacción es definir las secuencias de uso, establecer controles y sistemas de interfaz, accesibilidad, sistemas de orientación, estructura de navegación, niveles de acceso, tipos de acceso (seguridad) y utilización. De conformidad con el enfoque HCD una de las herramientas más efectivas para dirigir los esfuerzos de todos los involucrados en el proyecto es el llamado *storyboard*. Un *storyboard* es una serie de dibujos, diagramas y palabras que muestran exactamente lo que los usuarios verán y harán en cada momento (en cada pantalla), expone la funcionalidad del sistema y las relaciones temporales de los eventos. En este momento se usan los métodos descritos en el numeral 1.2. en cada iteración se validan los *storyboards* y *wireframes* con los usuarios finales para garantizar que el producto cumple con las expectativas de uso.

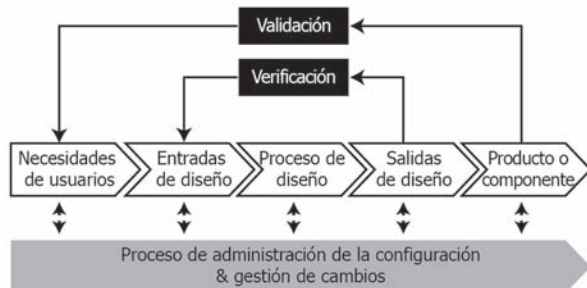
2.3 DISEÑO DE LA INTERFAZ DE USUARIO.

Diseñar las interfaces, definir los elementos de comunicación entre la herramienta y los usuarios,

principalmente aquellos que involucran la experiencia visual. Sin embargo, debe también integrar los demás elementos sensoriales involucrados en la experiencia de usuario como: oído, tacto, olfato y gusto. Una buena interfaz es un conjunto de elementos que encajan como bloques de construcción y que se pueden mover y reutilizar en muchas combinaciones diferentes y su relación familiar produce un sentido de continuidad y consistencia. El *storyboard* es el punto de partida para el diseño de la interfaz donde todos los elementos de ella existen de manera conceptual; después, el objetivo es traducir el *storyboard* en elementos de interacción con el usuario a través de un lenguaje sensorial, que se manifiesta a través de composiciones gráficas, sonidos, respuestas táctiles, etc.; es importante analizar el estilo y el diseño de una interfaz como un sistema unificado construido por partes (fondos, ventanas y paneles, botones y controles, iconos, imágenes, textos, videos, sonidos, animaciones, etc.).

Este enfoque metodológico se debe ejecutar de manera iterativa de tal forma que permita volver a las fases anteriores, permitiendo la evolución del producto en diferentes ciclos de acuerdo al grado de claridad de las necesidades de los usuarios, cuyas salidas del proceso de diseño son verificadas y los componentes finales son validados y revisados (ver Figura 2.). Por tanto, los métodos de evaluación son usados constantemente y los cambios propuestos al diseño son controlados a través de un proceso de administración de la configuración y gestión de cambios.

Figura 2. Proceso iterativo del modelo.



3. MODELO DE CICLO DE VIDA DEL SOFTWARE.

A pesar de que hoy en día las empresas informáticas conocen los beneficios que trae consigo incorporar estrategias de diseño e innovación en sus procesos, continúan utilizando modelos tradicionales, tal como lo afirma Kan en [23]. Para implementar el modelo descrito anteriormente como un proceso de diseño, en un contexto real, se tomó como unidad de análisis la empresa desarrolladora de software FCV.Soft,

Unidad Estratégica de Negocios (UEN) de la Fundación Cardiovascular de Colombia (FCV). El modelo que aquí se presenta se utilizó como base para construir el ciclo de vida de desarrollo de software de la organización. De esta manera se logra integrar los métodos de diseño descritos en la sección 1 y basados en el modelo descrito en la sección 2.

FIGURA 3. Metodología usada para la transformación de los procesos en la empresa.



Ahora bien, en un contexto real en una organización cuyos procesos están siendo controlados mediante un sistema de gestión de la calidad, fue necesario realizar la propuesta en el marco de una metodología de mejoramiento continuo. Es así que para la implementación de las modificaciones propuestas a los procesos organizacionales en FCV.Soft se utilizó el enfoque metodológico propuesto por Harrington [24] (Ver Figura 3) el cual consiste en las siguientes etapas: Etapa I: Preparar la organización para el mejoramiento (marco de referencia); Etapa II, analizar y entender los procesos (diagnóstico); Etapa III, propuesta de mejoramiento; Etapa IV, proponer indicadores y controles; Etapa V, Implementación de la propuesta. Con el modelo propuesto se llevaron a cabo tres proyectos de desarrollo de software con la participación de personal de la empresa FCV.Soft de acuerdo a requerimientos establecidos por la Fundación Cardiovascular de Colombia (ver Tabla 1):

- Historia Clínica Electrónica (HCE) módulo que hace parte del Sistema de Administración Hospitalaria Integrado (SAHI)
- Sistema Experto para soportar la toma de decisiones en Unidades de Cuidado Intensivo Neonatal (SE-UCI), funciona como un componente adicional para HCE.
- Portal Web para la FCV (Web-FCV).

Tabla 1. Descripción de los proyectos y participantes.

Nombre del proyecto	Participantes				
	Analistas	Ing. desarrolladores	Diseñador	Ing. calidad	Arq. de software
HCE	4	6	2	3	1
SE-UCI	2	2	1	2	1
Web-FCV	2	1	1	2	1

3.1 ETAPA I. MARCO DE REFERENCIA.

Para el diagnóstico contextual se examinaron las necesidades de la organización, los objetivos estratégicos del negocio y las principales características del sector de la empresa, en este caso particular, herramientas informáticas para el sector salud.

Se estableció un glosario con los principales conceptos, el lenguaje utilizado y un listado de abreviaturas en el contexto en el que se enmarcan los proyectos de la organización, con el ánimo de establecer un dialogo sin ambigüedades y confusiones en la comprensión de los proyectos.

3.2 ETAPA II. DIAGNÓSTICO.

En esta etapa se estudió el proceso de desarrollo usado en FCV.Soft. En detalle, desde la visión general hasta la identificación de sus atributos y/o vacíos con el objetivo de comprender su estado, flujos de información e involucrados, se hicieron cuestionarios y se esquematizó el proceso tal y como era entendido por los colaboradores de la empresa. Después de esta etapa se planearon las propuestas de mejora integrando a los responsables de conformidad con la realidad empresarial. Como método se usó el ciclo de Deming (planear, hacer, verificar, actuar) ya que es el método que usa la empresa para emprender acciones correctivas en su sistema de gestión de la calidad.

3.3 ETAPA III. PROPUESTA DE MEJORAMIENTO.

A partir del diagnóstico inicial se identificaron las áreas de mejora, sus problemas y principales causas; también se formularon los objetivos a lograr con la transformación del modelo. En esta etapa se incluyeron capacitaciones en las técnicas descritas en la sección 1 y todos los instrumentos y material necesario para conformar los métodos de diseño que se integraron dentro de los procesos de la organización, así como las actividades y procedimientos para su aplicación. Esta labor se

realizó con los jefes de área y el personal responsable de los procesos en FCV.Soft. El diseño del ciclo de vida se realizó sobre la base del modelo Proceso Unificado (RUP) ya que es un modelo adecuado y conocido en el ámbito local, como lo describe Claudia López en [25]. La propuesta no solo integra los métodos de diseño definidos en la sección 1, sino también las buenas prácticas para el desarrollo de software tales como: desarrollo iterativo incremental, administración de requerimientos, arquitectura basada en componentes, modelamiento visual, aseguramiento continuo de la calidad y control de cambios.

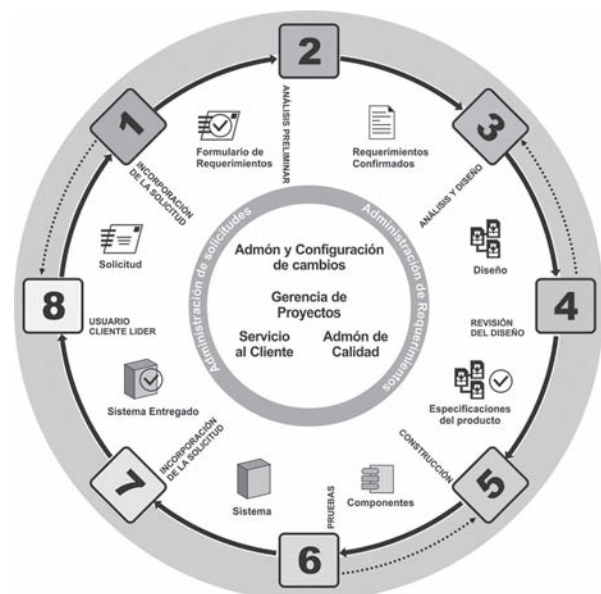
Con los jefes de áreas de la empresa se definieron los roles necesarios y se determinaron los cargos que de acuerdo a la estructura organizacional desempeñaría cada colaborador.

El modelo de ciclo de vida propuesto se compone pues de siete fases, como se muestra en la Figura 4:

3.3.1 Incorporación de solicitudes.

Recibir y comunicar las necesidades del cliente a la organización, recepción y clasificación de solicitudes, despacho de solicitudes. En esta parte se establecieron protocolos de comunicación entre los analistas y los usuarios finales, además de la implementación del proceso de administración de la configuración y gestión de cambios (CCM, por su sigla en inglés *Configuration and Change Management*).

FIGURA 4. Modelo de Ciclo de Vida de desarrollo de software propuesto para FCV.Soft.



Los roles líderes de este proceso son: el *analista de requerimientos* y el rol *administrador de la configuración y cambios*. Se propuso un ítem de entrada que consiste en un formulario donde se especifica el tipo de solicitud, se asignan los recursos necesarios, se establece un calendario preliminar y se especifica como realizar la trazabilidad de la solicitud. Este ítem es controlado por el proceso CCM.

3.3.2 Análisis preliminar.

Entender necesidades del cliente en términos de atributos del producto y requerimientos apropiados, viables y factibles. Se realizan actividades para capturar las necesidades, delimitar el sistema, representar los procesos, construir los diagramas de actividades UML, definir los casos de uso, identificar actores y establecer requerimientos. Se tiene en cuenta restricciones técnicas, económicas, legales y operativas y se hace la negociación para favorecer las expectativas tanto del cliente como de la empresa (modelo del negocio). Se realizaron instructivos y capacitaciones en los métodos descritos en la sección 1.1 y de acuerdo al modelo (diseño de la información). El rol líder es el analista de requerimientos, apoyado por el diseñador de interfaces para la ejecución de las diferentes aproximaciones con los usuarios finales, la clasificación de requerimientos se hizo con la estructura FURPS+ (Funcionalidad, Usabilidad, Confiabilidad, Desempeño, Soporte), propuesta en RUP. El principal ítem de salida es la visión del producto, las reglas del negocio, y los esquemas preliminares de representación de las actividades y procesos.

3.3.3 Análisis y definición.

Obtener requerimientos a un nivel de detalle suficiente para iniciar a planificar en el sistema las pruebas de evaluación. Todo requerimiento especificado describirá comportamientos externos del sistema perceptible por parte de los usuarios, para conseguir un entendimiento común de la aplicación a desarrollar entre el cliente y el proyecto. Se realizó entrenamiento y definieron instructivos en los métodos descritos en la sección 1.2. y de conformidad con lo definido en la sección 2.1 Diseño de la información y 2.2 Diseño de la interacción. La ejecución de estas actividades siguiendo los métodos del modelo permitieron una comunicación asertiva entre los usuarios finales y el equipo de desarrollo.

3.3.4 Aprobación del diseño.

Con las aprobaciones de diseño por parte de los usuarios finales, se garantizó que se estaba haciendo el producto correcto, de la manera adecuada. Es decir un ciclo constante de revisión - verificación - validación de la propuesta de diseño antes de decidir sobre una versión final del producto. Entonces, se brinda al cliente la posibilidad de evaluar el producto mostrándole las opciones de navegación e interacción mediante *stoyboards* y *wireframes*, para tomar las decisiones en cuanto a funcionalidad y comprobando así que el prototipo responde de la forma deseada por el cliente. Se ejecutan los métodos descritos en la sección 1.4 teniendo en cuenta los procesos de verificación y validación como se muestra en la Figura 2.

3.3.5 Construcción.

Producir componentes software que correspondan con la definición del producto y satisfaga los requerimientos, así como el conjunto de pruebas unitarias. Se implementan las funcionalidades especificadas mediante el lenguaje de programación elegido. Se realizan las ayudas en línea y los manuales de usuario. Una vez el producto está diseñado se construye el sistema de interacción y comunicación con los usuarios como se define en la sección 2.3

3.3.6 Verificación y validación.

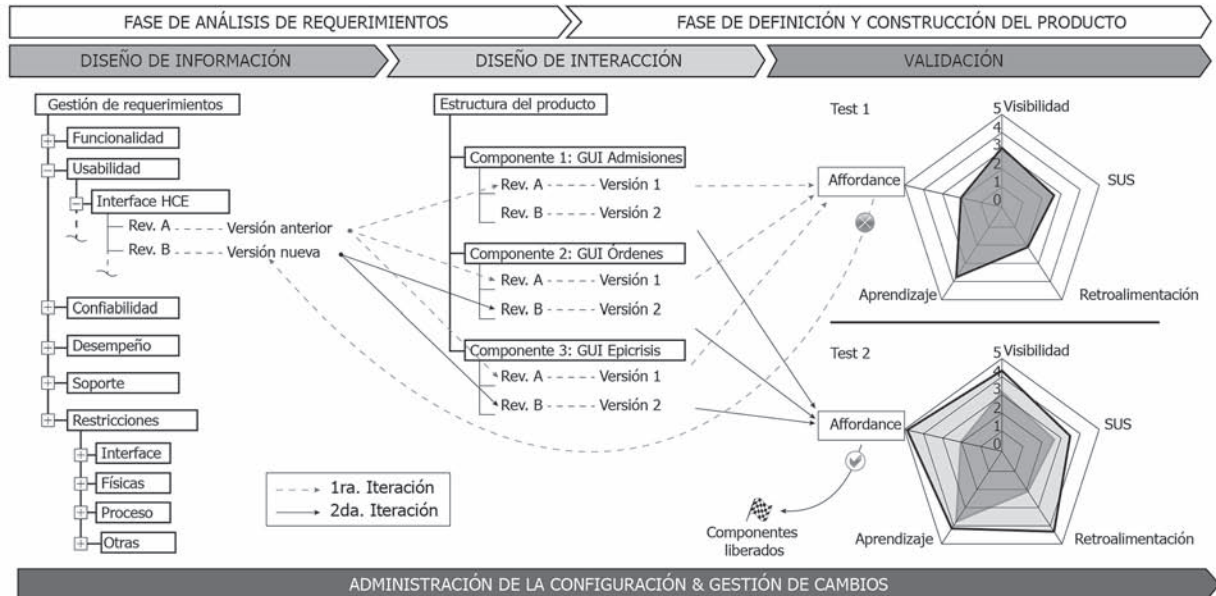
Se prueban los componentes software basados en el plan de calidad, y se verifica que el software cumple con los requerimientos los cuales se categorizaron según FURPS+. Se realiza el plan de pruebas, se especifican las pruebas a desarrollar, se ejecutan y se evalúan. Se realiza el reporte de los resultados. Este fue un proceso iterativo que permitió realizar una trazabilidad de los requerimientos y asociarlos a elementos específicos del producto, soportado en el proceso CCM. Para ilustrar este proceso en el modelo presentamos uno de las múltiples iteraciones en la evaluación de usabilidad de tres elementos de la GUI del producto HCE: Admisiones, órdenes y epicrisis (ver Figura 5.), en la figura se puede observar una iteración en la que un requerimiento de usabilidad es establecido, diseñado y luego validado mediante cinco variables (para este caso particular): visibilidad, retroalimentación, facilidad de aprendizaje y *affordances*². Además, se uso la escala SUS (System

²Capacidad que tienen los objetos de ser intuitivos y sugerentes para su uso mediante su aspecto formal en una situación determinada.

Usability Scale) como referencia general para tener una referencia cualitativa de la aceptación del sistema o del componente de interfaz evaluado. El control sobre el

requerimiento se da mediante la gestión del proceso de diseño a través del proceso CCM.

Figura 5. Ejemplo del modelo aplicado en HCE.



Algunas de las ventajas que se pudieron apreciar fueron: el poder comparar los resultados de la validación de los componentes de la interfaz de un producto contra los resultados que se habían obtenido en versiones anteriores. Además, de apreciar el impacto que el cambio a un componente tiene sobre el sistema u otros componentes.

3.3.7 Implantación.

Se planea y ejecuta la implantación del software garantizando la puesta en marcha del sistema bajo el uso de una metodología que aseguró el cumplimiento de los estándares de calidad de la empresa, las expectativas del cliente y la seguridad e integridad de la información. Se da asistencia a los usuarios (entrenamiento, capacitación, guía, soporte y documentación complementaria).

Como se muestra en la Figura 4 cada fase concluye con un hito del proyecto. De manera transversal y como centro del esquema se encuentran otras áreas de proceso adicionales, entre ellas: Administración de proyectos, Administración de la configuración, Servicio al cliente, Gestión de procesos y Aseguramiento de la calidad.

En cada fase se realizan actividades de verificación y validación, favoreciendo que la participación de los usuarios finales no se restrinja al inicio y finalización del proyecto, buscando su culminación exitosa basado siempre en el enfoque del HCD.

Estas fases comprenden un marco de trabajo general en el que se configuraron para el trabajo de cada uno de los proyectos ejecutados. En la parte izquierda de la Figura 4 se puede observar que el modelo se enfoca en la administración de solicitudes mientras en la derecha vemos un énfasis en la administración de requerimientos, es decir, la mitad de los procesos totales se definen en el ambiente del usuario y los procesos restantes se definen en el ambiente de la empresa.

Cada fase está compuesta por procedimientos y la sucesión de estas fases puede ampliarse con ciclos de retroalimentación con el usuario de la herramienta que se está diseñando. Es decir, una misma fase se puede ejecutar más de una vez a lo largo de un proyecto, de conformidad con lo establecido en el enfoque metodológico de diseño como proceso. Al final de cada fase el equipo de proyecto realiza una evaluación para determinar si los objetivos se cumplieron, y continuar a la fase siguiente.

3.4 ETAPA IV. MEDICIONES Y CONTROLES.

Se presenta el enfoque estructurado utilizado para proponer los mecanismos de control del plan de mejoramiento de los procesos de desarrollo de software en la empresa, que comprende el diseño de indicadores de eficiencia, eficacia y satisfacción de los clientes internos y externos de la empresa y se diseñan los mecanismos de control y seguimiento a los procesos. Además de los indicadores de los procesos, se incluyen como parte del sistema de calidad del producto métricas de usabilidad.

3.5 ETAPA VI. IMPLEMENTACIÓN DEL MODELO.

Se define el mapa de procesos de la organización, se realiza un inventario de procesos y su jerarquía dentro del mapa general y los métodos de desarrollo, la visión general de la empresa y las características del proceso de desarrollo; Se evalúa el proceso con asesores externos del equipo técnico de la empresa mediante los servicios de una empresa consultora y se hace el descubrimiento de hallazgos (generales, específicos y organizacionales) en cada una de las áreas de proceso de conformidad con el nivel III de madurez del modelo CMMI.

Se realizó un diagrama de los flujos de trabajo e información dentro de la organización para analizar los problemas y posibles soluciones. Después se establecieron los objetivos y alcances del modelo.

4. VALIDACIÓN DEL MODELO.

Para la validación del modelo se siguieron algunos lineamientos establecidos por la empresa consultora que evaluó los procesos. En particular este proyecto hizo parte de un proyecto para el logro de la implementación y cumplimiento de las prácticas de Nivel 3 del modelo CMMI patrocinado por USAID Colombia y Proexport. FCV.Soft hizo parte de este programa dirigido a Pymes colombianas y que es ampliamente descrito en detalle por Ricardo Llamosa en [26]. Por esta razón se toma CMMI como referencia para la evaluación del modelo de ciclo de vida de desarrollo propuesto. La implementación y validación de los procesos y procedimientos en su totalidad, se realizó con el desarrollo de proyectos ejecutados por la empresa (ver Tabla 1). CMMI exige que se demuestren evidencias concretas y verificables de por lo menos tres proyectos piloto reales para presentarse a la evaluación formal del modelo.

Se desarrollaron tres actividades principales: primero la adaptación de los procesos, que consiste en adecuar los procesos a las diferentes exigencias planteadas

en la organización. Segundo, la adaptación de las personas, que consiste en implantar los procesos, es decir, la utilización de los mismos por todas las personas involucradas. Finalmente, la evaluación, que permite ir midiendo el progreso de la implementación y verificar la institucionalización de las prácticas y la consecución de los objetivos de la mejora.

Para la evaluación final del modelo se utilizó el método SCAMPI (*Standard CMMI Appraisal Method for Process Improvement*) que está diseñado para proporcionar medidas de evaluación de la calidad en relación con los modelos de madurez, es decir, SCAMPI permite determinar el nivel de madurez de una empresa en relación al modelo CMMI y sólo lo puede realizar personal certificado, por lo cual se realizaron los cursos correspondientes en *Carnegie Mellon University* y se recibió la acreditación que otorga el *Software Engineering Institute* (SEI) para ser evaluador (*Team Member Appraisal*).

A continuación se presentan algunos de los resultados de la implementación del modelo de acuerdo a las metas propuestas y los medios para conseguirlo.

Meta 1: Realizar diagnóstico de los procesos de la empresa para el nivel III de madurez del modelo CMMI.

¿Cómo se logró?: Se utilizaron herramientas de análisis como la matriz DOFA, diagramas causa-efecto, y el método PMAM (Proceci Mini Assessment Method – Método usado por la empresa consultora), de esto se obtuvo una visión general de los procesos y las caracterizaciones de cada una de las áreas de proceso involucradas en el desarrollo de software en FCV.Soft.

Los resultados del diagnóstico permitieron identificar las fortalezas, puntos críticos y oportunidades de mejoramiento, así como las causas de las principales deficiencias en cada una de las áreas de mejoramiento.

Meta 2: Caracterizar los procesos a partir de un diagnóstico preliminar con base en el conocimiento de la empresa y el nivel III del modelo CMMI.

¿Cómo se logró?: Se realizó con una metodología de trabajo en equipo con participación del personal de la empresa en las propuestas de mejora y en el establecimiento del modelo propuesto para el ciclo de vida de desarrollo, las políticas, los procesos y los procedimientos.

Se definieron objetivos de la transformación, se definió el diseño del modelo para el ciclo de vida de desarrollo

de software de FCV.Soft., se establecieron políticas, procesos y procedimientos acordes a las necesidades de la organización, el diagnóstico realizado, la interacción con los empleados, el conocimiento recopilado y la aplicación del modelo CMMI.

Meta 3: Formular las estrategias para la transformación de los procesos de desarrollo de software en la empresa, con base en los aspectos metodológicos del diseño.

¿Cómo se logró?: Se formularon estrategias para vencer la resistencia al cambio y la identificación de los factores claves de éxito a tener en cuenta para la implementación del modelo, se capacitó a los involucrados en los diferentes métodos de diseño descritos anteriormente (ítem 1) y que harían parte del modelo.

Se diseñaron e implementaron estrategias de comunicación para el éxito de la implementación del modelo. Se diseñó un set de herramientas de comunicación del modelo del ciclo de vida diseñado y se llevaron a cabo las actividades de implementación de acuerdo al plan de acción.

Meta 4: Proponer los indicadores y mecanismos de control.

¿Cómo se logró?: Se diseñaron indicadores de gestión, indicadores de eficiencia, eficacia y satisfacción para cada uno de los procesos, gestionando las propuestas con la misma metodología aplicada para el modelo.

Se utilizaron indicadores de seguimiento al cumplimiento de las actividades de mejoramiento y al avance de los procesos con respecto al trabajo desarrollado y al modelo de referencia.

Estos indicadores se clasificaron en: indicadores de gestión, indicadores de cumplimiento de actividades e indicadores del estado de los procesos con respecto al modelo de referencia.

Meta 5: Mejorar la percepción positiva del diseño de los productos desarrollados en cuanto a su usabilidad (Tabla 2).

¿Cómo se logró?: Se establecieron métricas de efectividad, eficiencia y satisfacción para evaluar la usabilidad de los tres proyectos desarrollados basados en la norma ISO 9241-11. Y como se había especificado anteriormente, se utilizó la escala SUS para verificar los cambios en la percepción de los usuarios en comparación con versiones anteriores de los mismos productos. (Ver Tabla 2).

Tabla 2. Resultados de las mediciones con la escala SUS contra versiones anteriores de cada proyecto.

Proyecto	Valor SUS Versión anterior	Valor SUS Nuevo producto	Muestra (# de usuarios)
HCE	39%	57%	26
SE-UCI	27%	73%	14
Web-FCV	42%	56%	32

4.1 LECCIONES APRENDIDAS.

Los resultados obtenidos a partir de los proyectos desarrollados utilizando el modelo propuesto nos sugiere algunas reflexiones:

- Sin importar el grado de complejidad de un proyecto, este se ejecuta más fácilmente si se da más énfasis y se dedica más tiempo a las etapas tempranas como el análisis de necesidades y el desarrollo de la arquitectura de información del producto.
- Existe una relación directamente proporcional entre los tiempos de desarrollo y la especificación detallada de diseños que usaron modelamiento visual a través de *storyboards*.
- No es conveniente iniciar el diseño visual de la interfaz sin antes haber diseñado la interacción.
- La integración, la evaluación y la verificación de estándares se facilita si se cuenta con los resultados específicos del modelo principalmente: diagramas de flujo, *storyboards* y guías de interfaz.
- La evaluación del modelo utilizando CMMI como referencia fue un proceso largo y costoso, e implicó el esfuerzo tanto del personal de base de la compañía como el compromiso de sus directivos, para que fuese posible llevarlo a todos los niveles de la organización.

5. CONCLUSIONES.

Los diferentes métodos y técnicas que se usan para el desarrollo, producción y evaluación de productos tangibles se acoplan y se adaptan a los modelos de IS y permiten obtener productos de mejor calidad dotando a los profesionales de herramientas metodológicas para tener un mejor control sobre los procesos, además de lograr el aumento de la satisfacción del cliente permitiendo que los productos evolucionen rápidamente hacia versiones mejoradas, esto fue evidenciado en las evaluaciones de los tres proyectos desarrollados.

Los métodos de diseño fueron de gran utilidad para los miembros del equipo de desarrollo durante el entero ciclo, sin embargo se pudo apreciar que

funcionan mejor si están categorizados en grupos de acuerdo a la naturaleza de las técnicas que emplean (indagación, prototipado, test, etc.), el momento más conveniente para su aplicación de acuerdo a las fases de desarrollo (análisis, diseño, desarrollo y evaluación) y el conocimiento y complejidad que se tiene del problema a resolver.

Los métodos de diseño confieren un conocimiento profundo de las necesidades de los usuarios, lo que ayuda a entender mejor los problemas y llegar más temprano a soluciones efectivas. El basar el modelo en los procesos de HCD garantiza que los productos tengan en cuenta todas las consideraciones de los usuarios reales y no estén diseñados solamente con las apreciaciones exclusivas de quienes desarrollan los productos.

Integrar cada requerimiento a una parte concreta del sistema y después a una determinada evaluación de variables de usabilidad permitió tener un control sobre los aspectos en los cuales cada producto podía mejorar, lo que hizo que las evaluaciones se integraran como una parte natural del proceso de desarrollo del software.

La evaluación de la usabilidad de los productos no se debe establecer como una secuencia de tareas independientes, sino como un proceso iterativo que hace parte del trabajo de un equipo multidisciplinario, el cual debe contar el talento de diversas áreas y campos del conocimiento.

A futuro el modelo deberá evolucionar hacia una integración total de la información proveniente de todas las áreas de proceso para poder evaluar no solo los impactos que un cambio tiene sobre la interfaz, sino el impacto que tiene sobre otros aspectos tales como son los económicos y tecnológicos. La representación mediante diagramas de flujo de cada área en el modelo se deben especificar en detalle y mediante un lenguaje de modelado que permita su implementación p.ej. en sistemas PDM (Product Data Management) lo que permitirá la automatización del proceso de CCM. Y la integración en tiempo real del trabajo de cada uno de los miembros del equipo de desarrollo.

El ambiente que envuelve a las organizaciones de Tecnologías de la Información es dinámico, y se encuentra en continuo y acelerado movimiento exigiendo una adaptación al cambio rápida y eficaz, para sobrevivir y competir exitosamente en el mercado. En

esta investigación se pudo establecer la importancia del diseño y el papel que juega este en el vertiginoso ritmo del cambio, pues se pueden enfocar los objetivos de la organización con una mentalidad abierta a la creatividad, hacia una cultura organizacional que favorece la buenas iniciativas y traza el camino para hacer las cosas de una manera diferente y mejor.

6. REFERENCIAS.

- [1] Antonelli, P., & Museum of Modern Art (2011), *Talk to me : design and the communication between people and objects*, New York: Museum of Modern Art 207 p.
- [2] ISO, *ISO 9241-11 (1998) Ergonomics of human-system interaction- Guidance on usability*, International Standard Organization.
- [3] IEEE Xplore (2012), *International Workshop on Usability and Accessibility Focused Requirements Engineering UsARE*, IEEE.; Piscataway, N.J.
- [4] Vilbergdottir, S. et al. (2014), *Assessing the reliability, validity and acceptance of a classification scheme of usability problems (CUP)*. Journal of Systems and Software. 87(0): p. 18-37.
- [5] Albers, M., & Still B. (2011), *Usability of complex information systems evaluation of user interaction*, CRC Press,; Boca Raton, FL. p. 1 online resource (xxvii, 371 p.).
- [6] Martínez, J, et al. (2013), *El rol del diseñador en el desarrollo de software*. Revista Arquetipo, 5: p. 7-16.
- [7] Jaspers, M. et al. (2005), *The think aloud method: a guide to user interface design*. International Journal of Medical Informatics. 73(11-12): p. 781-795.
- [8] Sutcliffe, A. et al. (2006), *A method and advisor tool for multimedia user interface design*. International Journal of Human-Computer Studies. 64(4): p. 375-392.
- [9] Shneiderman, B., & Plaisant, C. (2009), *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley ed. Vol. 5 edition. 2009: Prentice Hall. 624.

- [10] Molina, A., & Giraldo W. et al. (2012), *CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems*. Advances in Engineering Software, 2012. 52(0): p. 10-29.
- [11] Calinescu, R. et al. (2010), *Foundations of computer software modeling, development, and verification of adaptive systems, 16th Monterey Workshop 2010*, Springer: Berlin; Heidelberg; New York. p. xii, 238 p.
- [12] Poelman, W. et al. (2008), *Design processes what architects & industrial designers can teach each other about managing the design process*, 2008, IOS Press: Amsterdam 122 p.
- [13] Mayhew, D. (2005), *Cost-justifying usability an update for internet age. Books 24x7 Inc. EEUU*.
- [14] IDEO. (2011), *Human-Centered Design Toolkit: An Open-Source Toolkit To Inspire New Solutions in the Developing World*. 3th. Edition ed: IDEO.
- [15] Nemeth, C. (2004) , *Human factors methods for design : making systems human-centered*, Boca Raton: CRC Press. xxiii, 392 p.
- [16] Cross, N. (2011), *Design Thinking: Understanding How Designers Think and Work*, Bloomsbury Academic, 192 p.
- [17] Acuña, S. et al. (2012) A HCI technique for improving requirements elicitation. Information and Software Technology, 54 (12), 1357-1357.
- [18] Otto, K. et al. (2001), *Product design: techniques in reverse engineering and new product development*, Upper Saddle River, NJ: Prentice Hall. xxi, 1071 p.
- [19] Chrissis, M. et al. (2011), *CMMI for Development: Guidelines for Process Integration and Product Improvement*. 3rd Edition: SEI Series in Software Engineering.
- [20] Marcus, A. (2011), *Design, user experience, and usability theory, methods, tools and practice: first International Conference, HCI International, Orlando, FL, USA, proceedings. Part I, in Lecture notes in computer science 67692011*, Springer.: Berlin ; New York. p. xxxi, 718 p.
- [21] Nielsen, J. et al. (1995), *Usability inspection methods*, New York: John Wiley & Sons. xxiv, 413 p.
- [22] Salvendy, G. (2012), *Handbook of Human Factors and Ergonomics*. 4 edition ed: Wiley.
- [23] Kan, S. et al. (2002), *Metrics and models in software quality engineering*: Addison-Wesley Longman Publishing Co., Inc.
- [24] Harrington, J. et al. (1997), *Business process improvement workbook: documentation, analysis, design, and management of business process improvement*, New York: McGraw-Hill. xix, 314 p.
- [25] López, C, et al. (2002), *Aplicación Del Proceso Unificado De Desarrollo De Software A Nivel Local*. Revista Gerencia Tecnológica Informática, 2002. 1(1).
- [26] LLamosa, R. & Estrada L. (2010), *Aprendizaje y aplicación del modelo CMMI - DEV en Pymes de software colombianas. La experiencia RCCS*. Revista Gerencia Tecnológica Informática, 2010. Vol. 9(24): p. 57 - 76.