

Universidad de las Ciencias Informáticas



Facultad 8

**“Aplicación del Modelo Ideal al Proceso de
Transferencia en Proyectos de Software
Educativo”**



Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autores: Yinet Pichardo Acosta

Heidy Toyos Lantigua

Tutor: Ing. Javier Heredia Ruiz

Ciudad de La Habana, Cuba

Junio, 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

<nombre autor> <nombre autor> <nombre tutor>

Firma del Autor Firma del Autor Firma del Tutor

Datos de Contacto

Nombre y Apellidos: Ing. Javier Heredia Ruiz.

Fecha de nacimiento: 5 de noviembre de 1983.

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 83110524587.

Correo: jheredia@uci.cu

Situación laboral: Profesor Adiestrado, Departamento de Sistema Digitales.

Institución: Universidad de las Ciencias Informáticas.

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Currículo

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI) desde julio del 2007. Al graduarse pasa a formar parte del claustro de profesores de la UCI, en la disciplina de Práctica Profesional impartiendo en la Facultad 8 el Segundo Perfil de Calidad del Software durante el primer semestre y en el segundo semestre pasa a formar parte del claustro de la facultad 10 en el departamento de Sistemas Digitales en la asignatura de Seguridad Informática. En el tiempo que lleva como graduado ha ejercido como miembro de tribunales de diferentes eventos desarrollados en la universidad. Se encuentra cursando el Diplomado de Software Libre el cual le tributará para la maestría. Forma parte del proyecto con el Consejo de Estado, específicamente en el de las Casas de Protocolo.



"Tan sólo hace falta una pequeña idea, para hacer un gran sueño realidad. "

Fidel Castro Ruz.

*A mis dos Teresas por ser
mi vida, mi apoyo, las luces
que me dieron fuerza para
luchar por mis sueños.*

Heidy

*A mi mamá por ser la
mejor madre del mundo, lo
más grande que tengo.*

*A mi esposo por ser mi
vida, por todo su amor y
dedicación.*

Yinet

Agradecimientos

A la Revolución y en especial a nuestro Comandante Fidel Castro Ruz por darnos la posibilidad de formarnos y graduarnos como Ingenieras en Ciencias Informáticas.

A nuestro tutor Javier Heredia Ruiz por toda su ayuda, por estar siempre dispuesto cuando lo necesitamos y darnos la fuerza suficiente para seguir adelante.

A Olga y Daismary por ser nuestras amigas y apoyarnos en todo momento.

A Zilenys porque a pesar de estar lejos fue quien nos incentivó a realizar esta investigación.

A Virgen Yuliet y Liset por ayudarnos en la terminación de este trabajo.

Al grupo 8306 que durante estos cinco años compartieron con nosotras y nos apoyaron para terminar esta carrera.

Heidy

A Michael por todo su amor, por enseñarme tanto en estos 5 años de mi vida, por estar a mi lado apoyándome y consentirme tanto.

A mi mamá Teresita por confiar siempre en mí, por brindarme su amor incondicional todos los días de mi vida, guiarme siempre por el mejor camino y dedicarme una vida entera.

A mi abuela Teresa por quererme tanto, por enseñarme cada día a ser una mejor persona y simplemente por darme la alegría de estar a su lado.

A mi abuelo Ciro por ser mi ejemplo a seguir, por su educación y por hacerme reír tanto con sus intervenciones oportunas.

A mi hermanita Mariam que tanto quiero, por haber nacido para colmarme con sus alegrías y su cariño, espero que se cumplan todos tus sueños.

A mi hermano Onys por hacerme feliz cada día con sus ocurrencias y ayudarme a ser mejor.

Yinet

A mi adorada mamá Ana, por ser la razón de mi existir, por su empeño en mi formación profesional, por el solo hecho de contar con su presencia, por el orgullo de tenerla. Gracias por todo mami, por creer siempre en mí. Estos son mis resultados, te los obsequio. Te amo con todo mi corazón.

A mi esposo Ángel por todo su amor, su paciencia y apoyo incondicional que tanto me ayudó para poder llegar a la meta. Por ser parte de mí y haber apreciado y cambiado mi vida. Porque siempre has estado animándome cuando las cosas no me han ido del todo bien. Te amo mi vida.

A mi hermana Yineba por su cariño un poco extraño pero que aprendí a entender.

A mis grandes amigas del Pre que más que amigas son como mis hermanas Maye, Leivis, Yenis, Raisy, Teresa, Vilniela, Elizabeth B, Elizabeth M., Dayana y Yelenys.

A mi hermana Aylén que a pesar de no estar a mi lado siempre llevo presente, por seguir mi camino, ojalá y todo te salga bien.

A mis queridas primas Yiyá, Yanet y Lisbet que con su sencillez y ternura saben ganarse el cariño de todos, por todo el apoyo y la fuerza que me han dado a pesar de estar lejos y ser siempre mi orgullo y ejemplo.

A mi tía Maira por su cariño y apoyo en los momentos difíciles.

A mis queridas amigas Yeni y Yadira por ser tan leales, alegres y estar pendientes de mi todos estos años.

A mi compañera de tesis, por ser mi amiga y ayudarme en la terminación de este trabajo.

A todos aquellos que confiaron en mi y me ayudaron en la realización de este sueño. Gracias.

A Mily, por ser mi amiga, mi hermana, mi familia, por todo su cariño y ayuda ilimitada, y a su novio Ernesto por junto a ella apoyarme durante todo este tiempo.

A Juanita, por tenerme como su hija y darme tanto cariño.

A mi prima Sacris por su ayuda incondicional, por ser mi ejemplo.

A mi abuela Martha, a mis tíos Rolando y Marilyn por el apoyo y el cariño que siempre me han dado. Por hacerme reír aun en los malos momentos.

A mi compañera de tesis por ser mi amiga y aguantarme, porque sin ella no hubiese sido posible terminar este trabajo.

A todos los que me ayudaron de una forma u otra a construir este sueño.

Gracias.

Resumen

La industria del software cubana ha ido evolucionando de manera significativa, proporcionando grandes beneficios económicos y tecnológicos al país. La Universidad de las Ciencias Informáticas (UCI) se dedica al desarrollo de software, centrándose principalmente en el Software Educativo (SWE), siendo este la línea base que definirá el trabajo.

Actualmente en la UCI existen varios proyectos de SWE que han implantado la Estrategia de Transferencia del Modelo de Factoría de Software; con respecto a esto se han evidenciado algunos problemas, debido a que no se cuenta con un modelo que mejore el proceso de transferencia que se está llevando a cabo en los proyectos.

Para darle solución a los problemas existentes en cualquier proyecto de SWE que implante este tipo de estrategia se presenta esta investigación, donde se propone la aplicación de un procedimiento basado en un modelo de mejora de procesos que guíe de manera eficiente la producción de SWE en la UCI, que permita mejorar la calidad de los procesos y finalmente del producto obtenido. Una vez estudiados los modelos de mejora de procesos más usados a nivel internacional, analizando las ventajas y desventajas que traen consigo con el propósito de seleccionar el modelo más adecuado y puntualizando las características principales de cada uno; se decide aplicar el Modelo Ideal para la mejora del Proceso de Transferencia del Modelo de Factoría de Software.

Palabras Claves

Modelo Ideal, Mejora de Procesos, Software Educativo, Estrategia de Transferencia, Factoría de Software.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica.....	5
1.1 Introducción.....	5
1.2 ¿Qué es un proceso de desarrollo de software?.....	5
1.2.1 Software.....	6
1.2.1.1 Software Libre.	6
1.2.2 Calidad del Software.	8
1.2.2.1 Calidad de Software Educativo.	9
1.3 ¿Qué es la mejora de procesos?	11
1.4 Modelo.....	13
1.4.1 Importancia de aplicar un Modelo.	13
1.4.2 Modelos de mejora de procesos.	14
1.5 Modelos para el mejoramiento de procesos de desarrollo de software.	15
1.5.1 Modelo Impact.	15
1.5.2 Modelo Mesopyme.....	17
1.5.3 Modelo Bootstrap.....	18
1.5.4 Modelo Spice (ISO/IEC 15400 parte 7).	20
1.5.5 Modelo Ideal.	23
1.5.6 Análisis comparativo de los Modelos.....	25
1.5.6.1 Conclusiones del estudio comparativo.....	26
1.6 Estrategia de Transferencia.....	27
1.6.1 Factoría de Software.....	28
1.6.1.1 Transferencia del Modelo de Factoría de Software.....	29

1.7 Conclusiones.....	30
Capítulo 2: Propuesta de Solución.....	31
2.1 Introducción.....	31
2.2 Características generales del Modelo Ideal.	31
2.2.1 Fase de Inicialización.....	32
2.2.2 Fase de Diagnóstico.	32
2.2.3 Fase de Establecimiento.....	33
2.2.4 Fase de Ejecución o Actuación.	34
2.2.5 Fase de Aprendizaje.	35
2.3 Proceso de Transferencia del Modelo de Factoría de Software en la UCI.	36
2.4 Propuesta para la Mejora del Proceso de Transferencia del Modelo de Factoría de Software en los proyectos de SWE.	36
2.4.1 Aplicación de la Fase de Inicialización al Proceso de Transferencia.	36
2.4.2 Aplicación de la Fase de Diagnóstico al Proceso de Transferencia.....	38
2.4.3 Aplicación de la Fase de Establecimiento al Proceso de Transferencia.	39
2.4.4 Aplicación de la Fase de Ejecución al Proceso de Transferencia.....	43
2.4.4.1 Gestión de Proyecto.	43
2.4.4.2 El Personal.	45
2.4.4.3 Bases Tecnológicas.....	52
2.4.4.4 Centro de Inteligencia.....	55
2.4.5 Aplicación de la Fase de Aprendizaje al Proceso de Transferencia.....	58
Capítulo 3: Validación de la Propuesta	60
3.1 Introducción.....	60
3.2 Guía para validar la propuesta.....	60
3.3 Conclusiones.....	66

Conclusiones	67
Recomendaciones	68
Referencias Bibliográficas	69
Bibliografía	73
Anexos	77
Anexo 1: Ejemplo de la portada del Cuaderno del Ingeniero.	77
Anexo 2: Cuaderno de Registro de Tiempo.	78
Anexo 3: Resumen Semanal de Actividades.	79
Anexo 4: Cuaderno de Registro de Defectos.....	80
Anexo 5: El Cuaderno de Trabajo.....	81
Anexo 6: Lista de compromisos.....	82
Anexo 7: El Plan del Proyecto.	83
Anexo 8: Estructura del curso de ActionScript.	83
Anexo 10: Estructura del curso de Dreamweaver / IDE.	85
Anexo 11: Estructura del curso de Macromedia Flash 8.	87
Anexo 12: Estructura del curso de CorelDRAW Graphics Suite.....	89
Anexo 13: Estructura del curso de Macromedia Fireworks 8.	90
Anexo 14: Estructura del curso de Gimp.	92
Anexo 15: Modelo para la recogida de información referente al peso de los criterios.	93
Anexo 16: Modelo para la recogida de información referente a la calificación de los criterios.	95
Glosario	98

Índice de Figuras

Fig. 1 Proceso de desarrollo de software.	5
Fig. 2 Proceso de desarrollo de software.	6
Fig. 3 Reducción de costos y aumento en la satisfacción del cliente, como indicadores primarios de la mejora del proceso.	11
Fig. 4 Modelo IMPACT-Actividades.....	16
Fig. 5 Actividades de Bootstrap.....	19
Fig. 6 Partes de la ISO 15400.	21
Fig. 7 Fases del Modelo Ideal.	24
Fig. 8 Proceso de Transferencia de un Modelo para aplicar en una Factoría.	29
Fig. 9 Fase de Inicialización.	32
Fig. 10 Fase de Diagnóstico.....	33
Fig. 11 Fase de Establecimiento.	34
Fig. 12 Fase de Ejecución.....	35
Fig. 13 Fase de Aprendizaje.	35
Fig. 14 Estructura del SQA.....	44
Fig. 15 Estructura Organizacional	47

Índice de Tablas

Tabla 1 Comparación de los Modelos	25
Tabla 2 Plan de Acción	41
Tabla 3 Resultado del trabajo de los expertos.....	61
Tabla 4 Cálculo de la Dispersión (S) para hallar la concordancia entre los expertos.....	63
Tabla 5 Cálculo de concordancia.	64
Tabla 6 Calificación de cada criterio.....	65

Introducción

El acelerado avance mundial de las Tecnologías de la Información y las Comunicaciones (TIC), trae consigo que la investigación, el desarrollo y la asimilación tecnológica jueguen un papel primordial para prever su evolución. Desde comienzos del siglo, la mayoría de los países han hecho progresos apreciables en casi todos los ámbitos del sector de las TIC. Al mismo tiempo, siguen existiendo importantes diferencias, ya que la difusión de estas es muy desigual entre países subdesarrollados y desarrollados.

La presencia de las Tecnologías de la Información (TI) en el aula, es cada día mayor; sin embargo, es un reto garantizar que se haga un uso adecuado de las mismas. Parte de este reto es contar con software de calidad. (DÍAZ et al. 2003).

La producción de software en Cuba se ha desarrollado de forma acelerada abarcando diferentes esferas del país entre las que se encuentra la educación.

Una tarea de gran prioridad para el estado cubano es el desarrollo de la industria de software, debido al alto aporte económico que se espera que posea, así como para el aseguramiento de un grupo importante de las actividades del país.

La situación actual y las perspectivas de la industria de software cubana están caracterizadas por el trabajo que se viene realizando en materia de capacitación y en la inserción tanto en el mercado nacional como internacional. Anticipar las mejores formas de interacción entre la universidad y la producción es una necesidad permanente. Esto es particularmente interesante, complejo y útil cuando se trata de una mercancía, como el software, que incide directamente en un amplio espectro de aplicaciones, que va desde los procesos más simples de la vida cotidiana hasta los más trascendentes del desarrollo económico, social y humano. (ESTRADA 2002).

Para lograr estos objetivos fue creada en el año 2002 la Universidad de la Ciencias Informáticas (UCI) como parte de la batalla de ideas cuya misión es formar profesionales comprometidos con la Revolución y muy preparados en Ciencias Informáticas. Con su surgimiento, se ha ganado una infraestructura productiva que combina el estudio con la producción de software.

En esta institución existen múltiples proyectos productivos destinados a la informatización de la sociedad cubana, la comercialización de software en el exterior, así como al desarrollo de Software Educativo (SWE), el cual será la línea base que definirá este trabajo.

Los nuevos avances tecnológicos y la creciente importancia de producir software, requieren que se trabaje en el mejoramiento continuo de la calidad de los procesos para lograr el correcto funcionamiento de los mismos.

La UCI se divide en diez facultades, cada una de ellas opera utilizando sus propios procedimientos, prácticas, herramientas, metodologías, hacen uso de varios modelos, pero cogen de estos lo que necesitan, no tienen definido un único modelo para desarrollar sus procesos, en algunos casos hay procesos que no se realizan o se realizan de diversas maneras incluso dentro de cada facultad.

En los proyectos dedicados a la producción de SWE que aplican la Estrategia de Transferencia del Modelo de Factoría de Software no cuentan con fundamentos sólidos que aseguren la calidad de los mismos debido a que no se aplica un modelo para el mejoramiento de procesos que permita evaluar dicha calidad, que guíe la planificación e implementación de iniciativas para mejorar estos procesos de transferencia.

Con vistas a lograr la mejora en los procesos de desarrollo de software existen varias opciones, entre las que se encuentra el Modelo Ideal.

Los principales aportes que se esperan con la aplicación de modelos como este son los de contribuir al mejoramiento de los procesos en proyectos de SWE, elevar la calidad del producto y además aumentar:

- La reducción de costos y tiempo.
- La posibilidad de reproducir éxitos en proyectos.
- El control sobre los riesgos de procesos.
- La confianza y satisfacción del cliente.

Ah raíz de la situación problemática antes planteada se deriva el siguiente **problema científico**: ¿Cómo guiar la planificación e implementación del Proceso de Transferencia del Modelo de Factoría de Software aplicando el Modelo Ideal?

Por tanto el **objeto de estudio** estará constituido por los procesos de desarrollo de SWE en la UCI.

Esto trae consigo que el **campo de acción** sea el Proceso de Transferencia del Modelo de Factoría de Software en los proyectos de SWE en la UCI.

Para de esta forma dar cumplimiento al **objetivo general** el cual se basa en aplicar el Modelo Ideal para la mejora del Proceso de Transferencia del Modelo de Factoría de Software en los proyectos de SWE en la UCI.

Para el desarrollo de la investigación se parte de la **idea a defender** que si se aplica el Modelo Ideal al Proceso de Transferencia del Modelo de Factoría de Software en los proyectos de SWE en la UCI sería posible eliminar las deficiencias existentes en el mismo.

En pos de cumplir con el objetivo planteado y resolver los problemas antes mencionados se proponen las siguientes **tareas**:

1. Caracterizar el Modelo Ideal y otros ya existentes.
2. Analizar la Estrategia de Transferencia del Modelo de Factoría de Software.
3. Aplicar el Modelo Ideal al Proceso de Transferencia del Modelo de Factoría de Software.
4. Validar la propuesta.

A continuación se expone como quedará estructurado el trabajo con una breve descripción de los capítulos que lo componen.

Capítulo 1. Fundamentación Teórica; el cual abordará conceptos relacionados con los procesos, modelos, la mejora de procesos, la Transferencia del Modelo de Factoría de Software. Además se caracterizan diversos modelos para el mejoramiento de procesos, y luego de un análisis de estos se selecciona el modelo a aplicar.

Capítulo 2. Solución Propuesta; en este se presenta la propuesta del Modelo Ideal para la mejora de los procesos de desarrollo de software, se describen cada una de las fases que compone el modelo, además se da una propuesta de solución para la aplicación en los proyectos de SWE en la UCI.

Capítulo 3. Validación de la propuesta; en este capítulo se determina la probabilidad de éxito que tiene la propuesta, para esto se realizan un conjunto de cálculos los cuales se detallan con el objetivo de lograr un mayor entendimiento.

Capítulo 1: Fundamentación Teórica

1

1.1 Introducción.

En este capítulo se tratarán varios conceptos relacionados con los procesos de desarrollo de software, así como su mejora. Se expondrán además características significativas referentes a la Estrategia de Transferencia del Modelo de Factoría de Software. Se describirán los modelos para el mejoramiento de procesos más representativos encontrados durante la investigación, luego se seleccionarán los aspectos que se tendrán en cuenta en la aplicación del modelo y se explica el porque de la selección.

1.2 ¿Qué es un proceso de desarrollo de software?

Según Jacobson “un proceso define **quién** está haciendo **qué**, **cuándo** y **cómo** para alcanzar un determinado objetivo. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas que el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible. (JACOBSON et al. 2000).

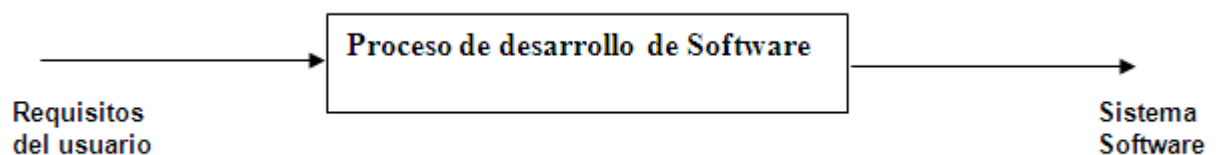


Fig. 1 Proceso de desarrollo de software.

Pressman caracterizando el proceso de desarrollo de software plantea que” se establece un marco común del proceso definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos del software, con independencia de su trabajo o complejidad. Un número de conjuntos de tareas que permiten que las actividades del marco de trabajo se adapten a las características del proyecto del software y a los requisitos del equipo

del proyecto. Finalmente las actividades de protección abarcan el modelo de procesos. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.” (S.PRESSMAN 2002).

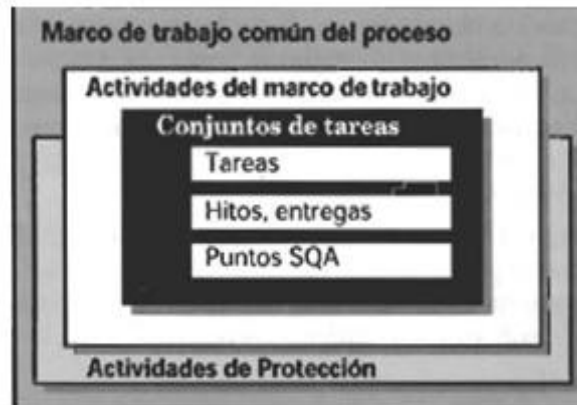


Fig. 2 Proceso de desarrollo de software.

De manera general, después de analizar los distintos conceptos sobre proceso dado por los autores se puede afirmar que un proceso de desarrollo de software es un grupo de actividades llevadas a cabo para desarrollar y gestionar sistemas de software, estas guían los esfuerzos de las personas implicadas en el proyecto; cada proceso tiene clientes y otras partes interesadas que son afectados por el proceso y quienes definen los resultados requeridos de acuerdo con sus necesidades y expectativas. Todos los procesos deberían estar alineados con los objetivos de la organización y diseñados para aportar valor, teniendo en cuenta el alcance y la complejidad de la organización.

1.2.1 Software.

1.2.1.1 Software Libre.

En la tesis de diploma, titulada “Desarrollo y Mantenimiento de una Distribución de Linux” los estudiantes González y Martínez afirman que el “Software libre es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente en Internet, o a precio del coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así y, aunque

conservar su carácter de libre, puede ser vendido comercialmente. Análogamente, el software gratuito (denominado usualmente Freeware) incluye en algunas ocasiones el código fuente; sin embargo, este tipo de software no es libre en el mismo sentido que el software libre, al menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.” (GONZÁLEZ and MARTÍNEZ 2007).

Galli plantea las siguientes ventajas del Software Libre:

- Independencia de productores y vendedores.
- Reducción de costos y mejor retorno de inversión a más del 99% de las empresas.
- Es un modelo eficiente de desarrollo.
- Respeto y apoyo a los estándares.
- Respeto al criterio científico: información de dominio público, revisión de los pares, proceso continuo de crítica, construcción incremental.
- Localización y regionalización de los beneficios. (GALLI 2004).

En síntesis, se pueden definir cuatro libertades de los usuarios del software entre las que se encuentran la libertad de usar el programa con cualquier propósito, de estudiar cómo funciona y adaptarlo a las necesidades de los usuarios. El acceso al código fuente es una condición previa para esto. Además la libertad de distribuir copias, de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

1.2.1.2 Software Propietario.

Según Montserrat el Software Propietario “se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, o que su código fuente no está disponible o el acceso a este se encuentra restringido. En el software no libre una persona física o jurídica posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades; de distribuir copias; o de mejorar el programa y hacer públicas las mejoras. De esta manera, un software sigue siendo no libre aún si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución.” (MONTSERRAT et al. 2006).

Ventajas del Software Propietario.

- La licencia incluye soporte técnico.
- Propiedad y decisión de uso del software por parte de la empresa.
- Soporte para todo tipo de hardware.
- El ocio para ordenadores personales está destinado al mercado propietario.
- Menor necesidad de técnicos especializados.
- Mayor mercado laboral actual.
- Mejor acabado de la mayoría de aplicaciones.
- Las aplicaciones número uno son propietarias.
- Mejor protección de las obras con copyright.
- Unificación de productos.
- Mayor difusión de uso. (SÁNCHEZ et al. 2006).

1.2.2 Calidad del Software.

Se puede considerar que la calidad del software es uno de los puntos de atención de las organizaciones actuales, ya que el software se ha convertido en un activo que determina en gran medida la operatividad de la organización.

Actualmente existen diferentes criterios sobre calidad. A lo largo del tiempo muchos han sido los que, de alguna forma u otra, han aportado nuevas ideas a esta rama, por lo que este concepto está matizado por diferentes puntos de vista. Algunos de los más conocidos y utilizados son:

Aiteco plantea que “es el conjunto de características de un producto que satisfacen las necesidades de los clientes y, en consecuencia, hacen satisfactorio al producto.” (Aiteco 2006).

Es la medida en que las propiedades de un bien o servicio cumplen con los requisitos establecidos en la norma o especificaciones técnicas, así como con las exigencias del usuario de dicho bien o servicio en cuanto a su funcionalidad, durabilidad y costo. (SEI 2002).

Analizados los conceptos anteriores se pueden apreciar elementos de gran importancia a tener en cuenta:

Como es el caso de las características del producto que deben satisfacer intereses y necesidades. Es por ello que todas las empresas productoras están haciendo lo posible por brindarles a sus clientes productos de máxima calidad. La producción de software no está eximida de esto. Existen ya algunas definiciones de calidad del software, tales como:

La Calidad del Software según Ferré “es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema a otro o de un programa a otro.” (Ferré 2005).

Pressman afirma que “la calidad se define como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.” (S.PRESSMAN 2002).

En resumen la calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. Esta puede medirse y varía de un producto a otro. Es multidimensional por lo que no se puede definir de forma simple. La noción de calidad implica que el producto desarrollado cumple su especificación.

1.2.2.1 Calidad de Software Educativo.

Antes de mencionar el tema de la Calidad del Software Educativo es importante conocer cuales son las particularidades que tiene este que hace que sea distinto de las vistas anteriormente, lo que conlleva a definir **Software Educativo**.

Pere Marquès señala que los Software Educativo son “programas educativos y didácticos como sinónimos para designar genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje.” (MARQUÈS 2005).

Según Cataldi el Software Educativo puede definirse como “los programas de computación realizados con la finalidad de ser utilizados como facilitadores del proceso de enseñanza y consecuentemente de aprendizaje, con algunas características particulares tales como: la

facilidad de uso, la interactividad y la posibilidad de personalización de la velocidad de los aprendizajes.” (CATALDI 2000).

Finalmente, luego de analizada la documentación existente puede decirse que el SWE es una aplicación informática destinada a facilitar y mejorar el proceso de enseñanza y aprendizaje, permitiendo desarrollar habilidades cognitivas en los hombres, constituyendo un material efectivo para el desarrollo educacional en la actualidad.

Características del SWE.

Los programas educativos pueden tratar las diferentes materias, de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

- Son materiales elaborados con una finalidad didáctica, como se desprende de la definición.
- Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- Son interactivos, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- Son fáciles de usar; los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un video, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer. (MARQUÈS 2005).

Seguidamente se dan varios conceptos de distintos autores sobre la Calidad del Software Educativo.

Para Pressman “cuando se habla de calidad de Software Educativo, se requiere un producto que satisfaga tanto las expectativas de los docentes como de los usuarios, a un menor costo, libre de defectos y cumpliendo con ciertas especificaciones.” (S.PRESSMAN 2002).

Gros afirma que “la calidad del Software Educativo está determinada no sólo por los aspectos técnicos del producto sino por el diseño pedagógico y los materiales de soporte. Este último aspecto es uno de los más problemáticos ya que existen pocos programas que ofrecen un soporte didáctico. La evaluación de Software Educativo se ha centrado tradicionalmente en dos momentos:

1. Durante su utilización real por los usuarios, para juzgar su eficiencia y los resultados que con él se obtienen.
2. Durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa.
” (Gros 1997).

1.3 ¿Qué es la mejora de procesos?

Según Moen “el mejoramiento de un proceso es el esfuerzo continuo para saber acerca del sistema de causas en un proceso y para usar este conocimiento en el cambio y mejora del proceso y de esa manera reducir su variación, complejidad y mejorar la satisfacción del cliente.” (Moen 1991).

La siguiente figura muestra la importancia del mejoramiento de los procesos de software.

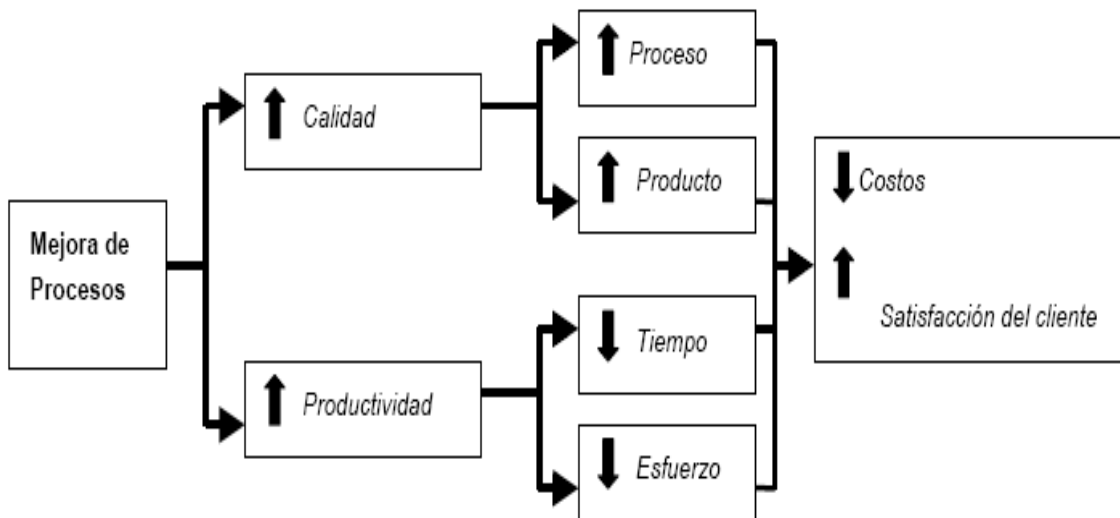


Fig. 3 Reducción de costos y aumento en la satisfacción del cliente, como indicadores primarios de la mejora del proceso.

Tantara opina que los modelos y estándares tienen un papel importante en la mejora del proceso de software pero cree que estos no son necesarios como requisito previo para la excelencia comercial. (Tantara 2001).

Sagrario señala que “la mejora de los procesos significa optimizar la efectividad y la eficiencia mejorando también los controles, reforzando los mecanismos internos para responder a las contingencias y las demandas de nuevos y futuros clientes. Es un reto para toda empresa de estructura tradicional y para sistemas jerárquicos convencionales.

Para mejorar los procesos, se tienen que considerar, entre otros, los siguientes aspectos:

- Análisis de los flujos de trabajo.
- Fijar los objetivos de satisfacción del cliente (tanto internos como externos), para dirigir la ejecución de los procesos.
- Desarrollar las actividades de mejora con los propietarios y actores del proceso.
- Responsabilizar e involucrar a los actores del proceso.” (SAGRARIO. 2005).

Mónaco y Vitali plantean que la mejora de procesos tiene como objetivos:

Aumentar la madurez de los procesos: grado en que están definidos, administrados, medidos y aplicados.

Aumentar la capacidad de los procesos: representa los resultados esperados que pueden ser obtenidos a partir de aplicar el proceso.

Para llevar a cabo un proyecto de mejora se requiere:

- Querer mejorar: está relacionado con las necesidades de la organización y la actitud del personal, la motivación y la personalidad de cada individuo.
- Poder mejorar: implica tener los medios necesarios y suficientes y contar con el conocimiento, experiencia y habilidad del trabajador, no solo para ejecutar bien sus tareas, sino también para ver las oportunidades de mejorarlas.
- Actuar en consecuencia: iniciar y llevar a cabo un proyecto de mejora de procesos.

Un proyecto de mejora de procesos exitoso requiere:

- Un patrocinador con poder de decisión: la Dirección de la organización está convencida y dispuesta a cambiar su historia.
- Sensibilización de toda la organización respecto de la Calidad y sus beneficios (todos deben ser y sentirse partícipes).
- Un modelo de referencia elegido, que sirva como hoja de ruta en el proceso. (MÓNACO and VITALI 2005).

En general se puede afirmar que la mejora de procesos es el esfuerzo planificado para obtener productos de mayor calidad. Suele basarse en modelos existentes, por ejemplo el IDEAL, que sirve como guía para la planificación e implementación de los procesos. Actualmente se cuenta con Mejoramiento de Procesos de Software (SPI), que no es más que un proceso para definir, gestionar, optimizar e institucionalizar los procesos de software de una empresa.

1.4 Modelo.

Jacobson plantea que “un modelo es una abstracción del sistema, especificando el sistema modelado desde en cierto punto de vista y un determinado nivel de abstracción. Un punto de vista, por ejemplo, una vista de especificación o una vista del sistema.” (JACOBSON et al. 2000).

Según Badillo “un modelo es una estructura conceptual que sugiere un marco de ideas para un conjunto de descripciones que de otra manera no podrían ser sistematizadas. El modelo cumple esta función en virtud de que une de manera inferencial, las proposiciones que afirman algo sobre los fenómenos que en él se integran. De esta manera, su estructura es diferente de la que se supone existe en el conjunto de fenómenos de la naturaleza.” (BADILLO 2004).

De los conceptos anteriores se puede considerar que un modelo se utiliza como ayuda para el pensamiento al organizar y clasificar conceptos confusos e inconsistentes. Es la representación concisa de una situación; por eso representa un medio de comunicación más eficiente y efectivo.

1.4.1 Importancia de aplicar un Modelo.

- Un modelo provee:
 - Un punto de partida.

- El beneficio de experiencias anteriores.
- Un lenguaje común y una visión compartida.
- Un marco de trabajo para priorizar acciones.
- Un modelo no es un proceso.
- El modelo muestra **Qué hacer**.
- El modelo da una guía sobre **Cómo hacerlo**.
- El modelo no indica **Quién debe hacerlo** ni **Con qué hacerlo**. (MÓNACO and VITALI 2005).

1.4.2 Modelos de mejora de procesos.

A continuación se exponen algunos conceptos y características sobre los modelos de mejora de procesos.

Para ayudar a las empresas en la mejora de sus procesos se crearon los modelos de mejora de procesos, ya que estos favorecen que la organización “ponga sobre la mesa” sus procesos actuales, reflexionen sobre ellos para comprender qué es lo que hace y por qué lo hace y en base a este estudio los optimicen para que estos sean lo más “eficaces” y “eficientes” como sea posible.

Ahora bien **¿Qué es un modelo de mejora de procesos?**

Es un conjunto estructurado de elementos cuyo objetivo es el desarrollo de productos de calidad de manera consistente y predecible...proporcionando a las organizaciones que los utilizan:

- Un punto donde comenzar.
- El beneficio de las experiencias de otras organizaciones.
- Un lenguaje común y una visión compartida.
- Un marco para priorizar acciones.
- Una forma de definir lo que significa “mejora” para la organización. (SAGRARIO. 2005).

En síntesis, un modelo de mejora se usa para priorizar y establecer objetivos de mejora, optimizando los procesos y proporcionando una guía que asegure el establecimiento de procesos estables, capaces y maduros.

1.5 Modelos para el mejoramiento de procesos de desarrollo de software.

En este epígrafe se definen características de los modelos de mejora de procesos más importantes encontrados durante la investigación, así como aspectos positivos y negativos de algunos de ellos. Varios de los modelos en cuestión han sido adoptados con éxito por empresas desarrolladoras de software. Los modelos antes mencionados son los siguientes:

- Modelo Impact.
- Modelo Mesopyme.
- Modelo Bootstrap.
- Modelo Spice (ISO/IEC 15400 parte 7).
- Modelo Ideal.

1.5.1 Modelo Impact.

Scott señaló que “Impact es un framework que propone un paradigma liviano para SPI, el cuál consta de los siguientes estados: Entender-Mejorar-Aplicar-Medir, los cuales pueden ser aplicados incrementalmente a lo largo de varios proyectos. Por otra parte, al implementar iteraciones sucesivas, las metas de mejoramiento serán alcanzadas rápidamente (usualmente dentro de unos pocos meses) y son factibles de medirse a través de los proyectos en los cuales el enfoque de mejoramiento ha sido aplicado. Éste framework diferencia dos niveles: el nivel de proyecto y el nivel de proceso. En el nivel de proyecto, muchos proyectos se desarrollan de acuerdo a buenas prácticas de gestión de proyectos (Por ejemplo Plan-Do-Check-Act (Planear-Hacer-Chequear- Actuar)). En el nivel de proceso, la experiencia y el entendimiento de muchos proyectos se usan para entender y mejorar el modelo de procesos genérico, el cual es usado después para guiar proyectos futuros, este es el ciclo que se llama el ciclo del proceso. Estos ciclos interactúan muy de cerca, el ciclo de proceso conduce la ejecución de los proyectos y los proyectos conducen las mejoras para el proceso.” (Scott 2001).

Francisco Pino enmarcó que “el framework de Impact se basa en que el proceso es una abstracción de las prácticas llevadas a cabo en muchos proyectos diferentes por muchas personas diferentes, así es posible aprender continuamente y mejorar el proceso desde la experiencia ganada por mucha gente en muchos proyectos.” (PINO et al. 2007).

Las actividades del Modelo Impact son:

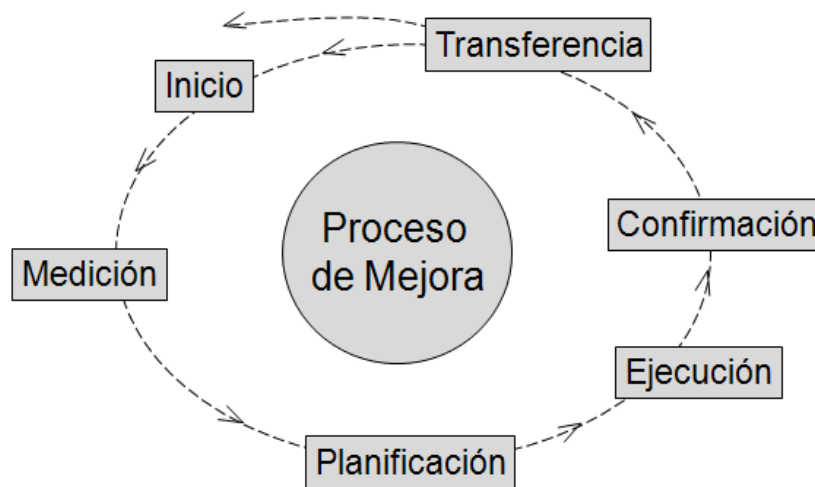


Fig. 4 Modelo IMPACT-Actividades.

Inicio

- Descubrimiento de los objetivos de negocio que soportan la mejora.
- Identificar el contexto organizativo de la mejora.
- Establecen el sponsor y el líder.
- Definir costos y beneficios.
- Establecer infraestructura y recursos.

Medición

- Definir el alcance de la evaluación.
- Planificar la evaluación.
- Ejecutar la evaluación.
- Desarrollar el informe de resultados.

- Presentar los hallazgos y recomendaciones.

Planificación

- Asociar la mejora a algún objetivo de la organización.
- Reconciliar iniciativas de mejora.
- Priorizar las iniciativas.
- Transformar objetivos genéricos de la mejora en objetivos medibles.

Ejecución

- Desarrollar soluciones.
- Desarrollar pruebas pilotos y probar la mejora.
- Instalar e implementar soluciones.

Confirmación

- Evaluar la mejora lograda.
- Determinar el alineamiento con los objetivos de negocio e impacto en los resultados.
- Proveer realimentación a los grupos de mejora y a la organización.

Transferencia

- Informar lecciones aprendidas para mejorar el proceso de mejora.
- Definir nuevas mejoras.

1.5.2 Modelo Mesopyme.

Mesopyme proporciona un método de mejora formado por fases y un paquete de actuación, que es un conjunto de componentes organizativos, técnicos y de gestión que ayudan a dar una solución concreta a un problema de software bien definido. Esta arquitectura proporciona una guía para soportar, analizar, diseñar y gestionar los procesos de software, de forma que sean consistentes con CMM, ISO/IEC 15504 o el modelo de procesos de referencia elegido. (SANZ and CUADRADO-GALLEGO 2005).

Según José A. Calvo-Manzano “el paquete de acción está estructurado por un conjunto de componentes que ayudan a obtener una solución específica para un dominio de proceso de software. El paquete de acción es el mecanismo de ayuda más rápido y asequible para SPI.” (A.CALVO-MANZANO et al. 2006).

Antonia Mas Pichaco expone que el método de mejora está formado por 5 fases:

- **Fase 1: Compromiso de la alta dirección.**
- **Fase 2: Evaluación.** El objetivo de esta fase es descubrir el estado real de los procesos de software de la empresa, es decir, obtener sus puntos fuertes y débiles, utilizando como referencia de modelo de proceso al CMM.
- **Fase 3: Solución de mejora.** Teniendo en cuenta las mejoras que se seleccionaron en la fase anterior, se propone una solución de mejora, que se desarrollará mediante la implantación de un grupo de productos base de mejora, denominado paquete de actuación y que da respuesta a las necesidades de un área clave de proceso, según CMM, o de un dominio de mejora, según otras aproximaciones.
- **Fase 4: Institucionalización.** El personal de la propia empresa implantará el paquete de actuación en todos los proyectos, consiguiendo así su institucionalización.
- **Fase 5: Refinamiento de la implantación.** (PICHACO 2007b).

Finalmente se puede alegar que una característica principal de este modelo es que se centra en reducir el tiempo y el esfuerzo en la implementación de SPI, basado en el concepto de paquetes de acción. Dicho modelo solo puede ser utilizado por organizaciones pequeñas y medianas.

1.5.3 Modelo Bootstrap.

El Modelo Bootstrap propone un método y los instrumentos necesarios que permiten identificar los puntos débiles de la organización, además de presentar los cambios necesarios para obtener una mejora de la situación. La idea que resumiría el concepto Bootstrap es la mejora cíclica propia de la filosofía Kaizen que propone una continua secuencia del tipo Planificar-Hacer-Comprobar-Acción. Como máxima fundamental de este modelo se puede indicar que la forma de superar la crisis del software es mejorando el proceso de organización por el cual el software es creado, manejado y mantenido. En este modelo la asunción básica es que el

desarrollo software puede ser presentado como un flujo lineal de actividades bien definidas que conforman el ciclo de vida. Una medida del proceso basada en un modelo de referencia y su escala. La medida se entiende aquí como una comparación con una escala ordinal. La escala de medida propuesta por el Instituto de Ingeniería de Software (SEI) a través de su modelo CMM fue aceptada como adecuada. (ESCORIAL 2007).

Antonia Mas Pichaco expresó que “el modelo de procesos de software utilizado por Bootstrap es un modelo cíclico, compatible con el que se utiliza en ISO/IEC 15504, que también es equivalente al modelo IDEAL del SEI. La metodología propuesta por Bootstrap comprende las actividades que se muestran en la figura.

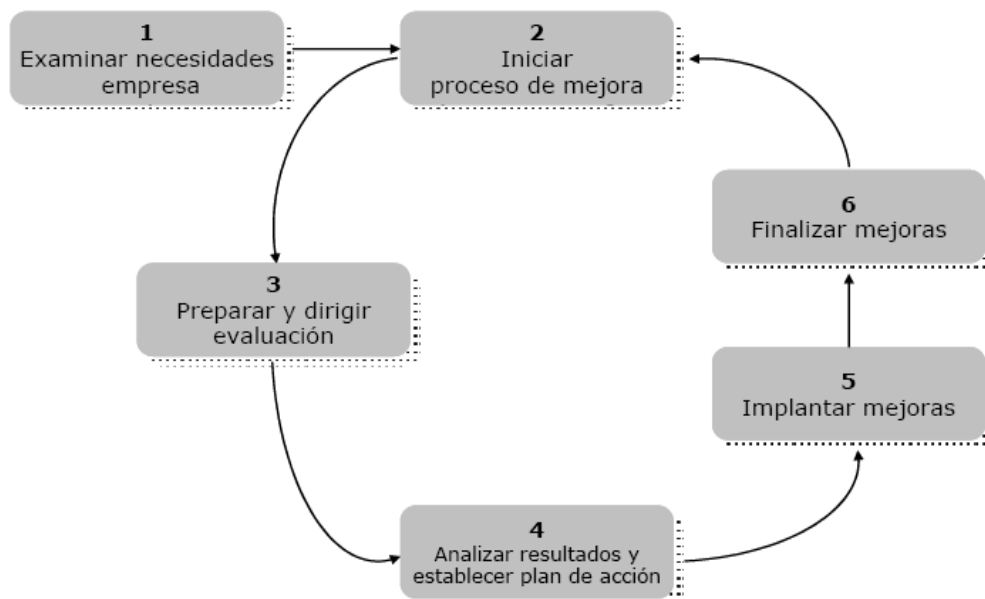


Fig. 5 Actividades de Bootstrap.

- **Examinar las necesidades de la empresa**, realizando los primeros contactos con los responsables y delimitando la acción.
- **Iniciar el proceso de mejora**, definiendo el objetivo de la mejora y comenzando a desarrollar un plan de mejora.
- **Preparar y dirigir la evaluación**, para conocer el estado de todos los procesos de la empresa.

- **Analizar los resultados de la evaluación y establecer el plan de acción**, identificando y priorizando las áreas de mejora y estableciendo el plan de acción para mejorar estos procesos seleccionados.
- **Implantar las mejoras**, a través de un plan de acción detallado aplicado a proyectos reales.
- **Finalizar las mejoras**, garantizando que se han alcanzado todos los objetivos establecidos.” (PICHACO 2007a).

De las características anteriores se puede considerar que entre las principales ventajas de este modelo está que genera una imagen de la organización en poco tiempo, mostrando sus fortalezas y debilidades. Además realiza evaluaciones a varios niveles, presenta mecanismos concretos para las mejoras, recomendaciones y las monitorea; utiliza y propone métodos probados para implementar prácticas de punta de Ingeniería de Software.

1.5.4 Modelo Spice (ISO/IEC 15400 parte 7).

En síntesis se puede plantear que la Organización Internacional para la Estandarización (ISO) en conjunto con la Comisión Internacional Electrotécnica (IEC) crearon ISO/IEC 15400, el cual provee un modelo para la evaluación, validación y optimización del proceso de desarrollo de software. Se centra en evaluar el proceso de acuerdo a niveles definidos, así como gestionar la calidad del mismo identificando las áreas de mejora y riesgos potenciales.

Según Antonia Mas Pichaco “ISO/IEC 15504 es un estándar internacional que es aplicable a cualquier organización que quiera conocer y mejorar la capacidad de sus procesos, independientemente del tipo de organización, del modelo del ciclo de vida adoptado, de la metodología de desarrollo y de la tecnología utilizada. ISO/IEC 15504 proporciona una base para realizar evaluaciones de la capacidad de los procesos de software y permite reflejar los resultados obtenidos sobre una escala común, que puede usarse por una parte para comprobar la evolución de una organización en el tiempo o para observar su situación respecto a la competencia, y por la otra, para la definición de estrategias de mejora. El estándar no pretende fijar la manera de realizar los procesos dentro de una organización, sino que valora su capacidad y ayuda a proponer mejoras que aumenten esta capacidad. La manera de llevar a cabo estas mejoras no entra dentro del alcance de la norma.” (PICHACO 2007a).

Jorge Salamanca indicó que el modelo ISO/IEC 15504 también está ideado para determinar la idoneidad de los procesos de otras organizaciones para un contrato determinado. Las partes de la norma ISO 15504 liberadas van desde la 15504-1 a la 15504-9.

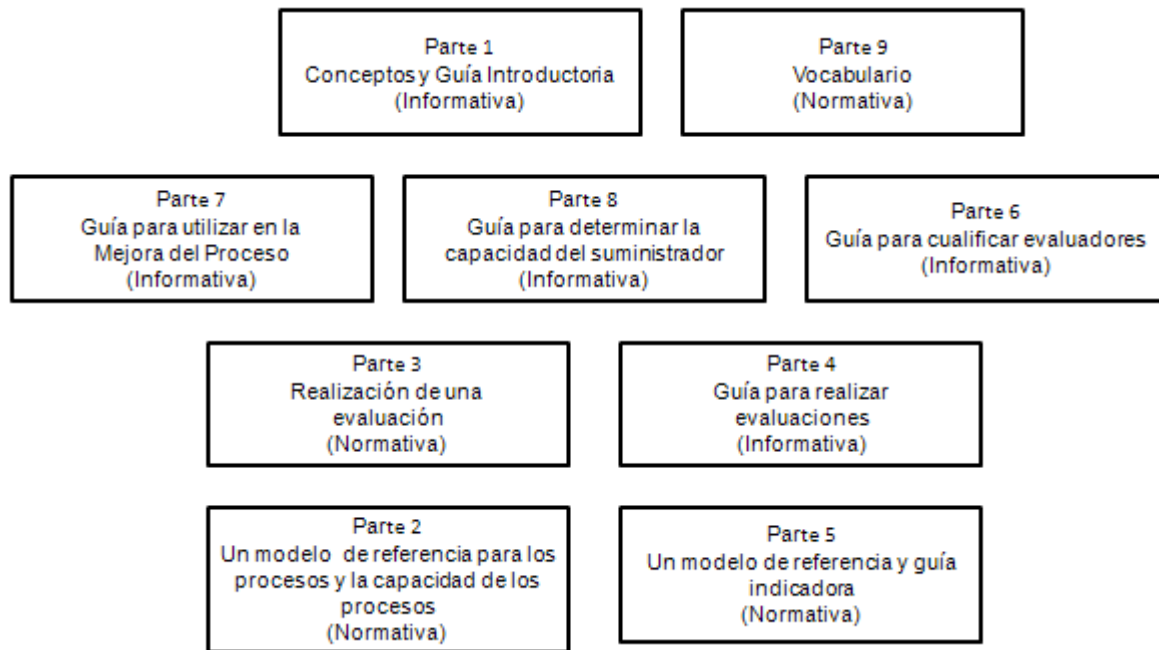


Fig. 6 Partes de la ISO 15400.

Los niveles considerados son:

- **Incompleto:** Hay un fallo generalizado al alcanzar los propósitos del proceso.
- **Realizado:** El propósito del proceso es generalmente alcanzado. Este éxito no tiene por que haber sido rigurosamente planificado ni seguido.
- **Gestionado:** El proceso libera productos de acuerdo a procedimientos específicos y es planificado y seguido.
- **Establecido:** El proceso es llevado a cabo usando procesos definidos basados en principios de la Ingeniería de Software.

- **Predecible:** El proceso definido es ejecutado en consistencia con controles de límites establecidos, para alcanzar los objetivos definidos. Las medidas detalladas del rendimiento son coleccionadas y analizadas.
- **Optimizado:** La realización de los procesos se encuentra optimizada de forma que coincidan con las necesidades actuales y futuras de negocio. Los resultados de los procesos son alcanzados de forma repetida de acuerdo con los objetivos definidos. (ESCORIAL 2007).

En su parte 7 reside este estándar internacional desarrollado en el proyecto Spice para la evaluación de proceso del software. El estándar ISO/IEC 15504 proporciona un marco para todos los aspectos de una evaluación de proceso que se puede utilizar para evaluar la capacidad de los procesos de su organización. El marco precisa los requerimientos para la realización de una evaluación conforme a la ISO/IEC 15504. (HURTADO 2005).

Además Marko Pyhäjärvi citó que “el documento 7 actúa como una guía sobre el uso del proceso de evaluación para entender el estado actual de los procesos, para crear y dar prioridad a los planes de mejora. El documento está destinado principalmente a la gestión de una organización, considerando un programa de mejora de procesos de software, los miembros de los equipos de mejora, ingenieros de software, y los consultores externos.

Esta guía usada en la evaluación para el mejoramiento de procesos, se ocupa de lo siguiente:

- Apelando a una valoración de proceso del software.
- Usando los resultados de una valoración del proceso de software.
- Midiendo la efectividad del proceso de software y la efectividad de las mejoras.
- Identificando acciones de mejoras alineadas con las metas del negocio.
- Usando el modelo de proceso como un mapa para las mejoras.
- Asuntos culturales en el contexto de las mejoras del proceso de software.
- Tratando con los asuntos de gestión para la mejora del proceso de software.” (PYHÄJÄRVI 2004).

En resumen, después de la investigación realizada se puede decir que este modelo es una base para realizar una evaluación detallada del estado actual del proceso de una organización.

Contiene un alto nivel de descomposición de sus procesos e indicadores, además de facilitar evaluaciones objetivas y con posibles resultados cuando es realizada por evaluadores altamente calificados. Como desventaja presenta que se requiere de un gran esfuerzo para realizar las evaluaciones y por lo tanto un alto costo.

1.5.5 Modelo Ideal.

Este modelo fue desarrollado por el SEI, es una guía para iniciar, planificar e implementar tareas para mejorar los procesos de desarrollo de software a través de diferentes fases y pasos que han de seguirse para lograr un trabajo efectivo en este ámbito.

Según Rincón “el Modelo IDEAL es llamado de esta manera, por las diferentes fases de trabajo que involucra: **Initiating**, **Diagnosis**, **Establishing**, **Acting** y **Leverageg**. Cada una de estas fases consta de diferentes actividades o pasos por desarrollar. Hay que tener en cuenta, que el trabajo realizado por medio de esta metodología contempla lo anteriormente dicho, manejar el mejoramiento progresivamente. El modelo está diseñado como un modelo cíclico, en el cual una vez se han terminado de ejecutar las diferentes actividades que este involucra, se debe volver a iniciar el ciclo de mejoramiento, llevando a cabo las mismas actividades pero definiendo nuevos objetivos para el ciclo que comienza.” (RINCÓN and RODRÍGUEZ 2006).

Gremla indicó que “IDEAL proporciona un acercamiento usable, comprensible a la mejora continua contorneando los pasos necesarios para establecer un programa de mejora acertado. Después de las fases, las actividades, y los principios del modelo IDEAL se han logrado beneficios en muchos esfuerzos de la mejora. El modelo proporciona un acercamiento disciplinado de la ingeniería para la mejora y establece las pautas para una estrategia a largo plazo de la mejora. El modelo consiste en cinco fases:

I –Iniciar: Sentar las bases para el éxito de los esfuerzos de mejora.

D-Diagnosticar: Determinar dónde se encuentra y dónde se quiere llegar.

E-Establecer: Planificación de los detalles de cómo llegar al destino.

A-Actuar Hacer el trabajo de acuerdo con el plan.

L-Aprender Aprender de la experiencia y mejorar la capacidad para adoptar nuevas tecnologías en el futuro.” (GREMBA and MYERS 1997).

Mónaco y Vitali describen las fases del modelo como se muestra a continuación:

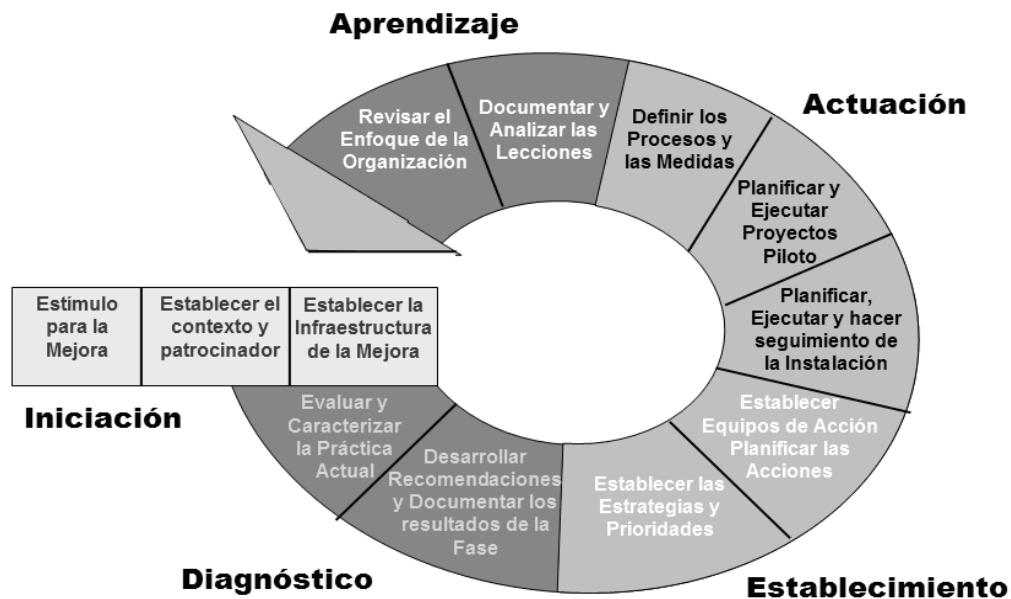


Fig. 7 Fases del Modelo Ideal.

Para McFeeley “el Modelo IDEAL, como otras aproximaciones para la introducción de mejoras en el proceso productivo de una organización tienen como propósito establecer los mecanismos para facilitar la realización de programas de mejora continua en organizaciones, aunque el Modelo IDEAL está especialmente orientado a las mejoras en organizaciones intensivas en software.” (McFeeley 1996).

A modo de conclusión se puede plantear que este modelo recoge muchas de las características de los modelos anteriores, además de que puede ser aplicado a cualquier organización. Es un modelo de ciclo vital para la mejora del proceso de software que sigue las mejores prácticas recomendadas del modelo CMM. Provee un enfoque disciplinado de ingeniería para SPI, focaliza en el gerenciamiento del programa de mejoras y establece los fundamentos para una estrategia de largo plazo. Además de que es un proceso cíclico, pues después de la fase de aprendizaje se puede volver a empezar para diagnosticar nuevas áreas mejorables y así lograr acercarnos a la perfección en el proceso de desarrollo de software que se desea.

1.5.6 Análisis comparativo de los Modelos.

En el análisis de los modelos para el mejoramiento de los procesos de desarrollo de software vistos anteriormente y teniendo en cuenta las características necesarias para mejorar el Proceso de Transferencia del Modelo de Factoría de Software, se seleccionó el modelo que se aplicará para dicha mejora. En la siguiente tabla se establece una comparación entre estos modelos, representando las características más significativas de cada uno.

Tabla 1 Comparación de los Modelos

Aspectos/ Modelos	Impact	Mesopyme	Bootstrap	Spice(ISO/IEC 15400 parte 7)	Ideal
Usa Referencias	No menciona nada al respecto.	CMM	ISO/IEC 12207 SPICE	Basado en SPICE	CMM
Aspectos a favor	Las metas del mejoramiento son alcanzadas rápidamente.	Se centra en reducir el tiempo y el esfuerzo en la implementación de SPI, basado en el concepto de paquetes de acción.	Genera una imagen de la organización en poco tiempo, mostrando sus fortalezas y debilidades. Utiliza y propone métodos probados para implementar prácticas de punta de	Ofrece una base para la evaluación muy detallada del estado actual del proceso de una organización. Más consensuado y probado.	Proceso cíclico, provee un enfoque disciplinado de la ingeniería para SPI, se focaliza en el gerenciamiento del programa de mejoras y establece los fundamentos para una estrategia de largo plazo. Consta de gran

			Ingeniería de Software.		prestigio internacional.
Aplicación	Pequeñas y Medianas Empresas.	Pequeñas y Medianas Empresas.	Cualquier Organización.	Cualquier Organización.	Cualquier Organización.
Niveles y/o Fases Actividades	6 actividades bien definidas.	5 fases bien definidas.	6 niveles bien definidos.	6 niveles bien definidos.	5 fases bien definidas.

1.5.6.1 Conclusiones del estudio comparativo.

El Modelo Impact propone un paradigma liviano para SPI, implementa iteraciones sucesivas por lo que las metas de mejoramiento serán alcanzadas rápidamente y son factibles de medir a través de los proyectos en los cuales el enfoque de mejoramiento ha sido aplicado. Pero por otro lado tiene como algo negativo que solo puede ser aplicado a pequeñas y medianas empresas al igual que el Modelo Mesopyme.

Con respecto al Modelo Bootstrap, este propone un método e instrumentos necesarios que permiten identificar los puntos débiles de la organización, además de presentar los cambios necesarios para obtener una mejora de la situación.

Por otro lado el Spice (ISO/IEC15400 parte 7) realiza una evaluación detallada del estado actual del proceso de una organización. Provee evaluaciones objetivas, con resultados repetibles y no necesitan abordarse a toda la organización, estas pueden realizarse únicamente en aquellos procesos que sean áreas de problemas. Una de sus desventajas es que se necesita un gran esfuerzo para realizar las evaluaciones y a su vez un alto coste pues son muy difícil de entender.

De manera general luego del análisis realizado se puede decir que el Modelo Ideal recoge muchas de las características de los modelos anteriores, además está basado en CMM y

consta de un gran prestigio a nivel internacional. Proporciona un enfoque usable y entendible a la mejora continua, guiando los pasos necesarios para establecer un programa de mejora exitoso. Es un proceso cíclico, ya que permite después de haberse implementado la fase final volver a empezar en busca de nuevas mejoras.

1.6 Estrategia de Transferencia.

Antes de llegar a la temática de Estrategia de Transferencia para un mayor entendimiento se cree importante definir **¿Qué es una Estrategia?**

La estrategia es tratar de entender cuál es tu posición actual... y evaluar con toda la inteligencia disponible, los cambios competitivos, los cambios en los mercados que se pudieran capitalizar o de los que nos debemos cuidar para ir de la posición actual a la posición deseada. Es evaluar en forma realista las posibilidades que tenemos de partir de aquí y llegar allá.” (WELCH 2006).

“Una estrategia es un modelo o un plan que integra las metas más importantes de la compañía con sus políticas y con sus acciones. Una estrategia bien formulada le ayuda a las empresas a conjuntar y dirigir sus recursos hacia una postura estratégica viable la cual descansa en las competencias y habilidades de la misma, en los cambios que se prevén en el medio ambiente y en las estrategias de sus competidores.” (QUINN 2006).

En el diccionario de la lengua española se encuentran dos definiciones de estrategia. La primera es referente al surgimiento de la palabra y se refiere al “arte de planear y dirigir las operaciones bélicas o militares: estrategia de defensa, de asedio”; mientras que la otra definición es más actual y se define como la “técnica y conjunto de actividades destinadas a conseguir un objetivo: estrategia de venta, electoral.” (ESPAÑOLA 2005).

Después de una revisión de la documentación sobre estrategia, se puede afirmar de manera general que es un patrón de decisiones coherente, unificador e integrador.

Estrategia de Transferencia.

Guerra y González plantean que “especialistas de la Universidad del Norte en Colombia, hicieron un estudio sobre las estrategias de transferencia, donde identificaron dos tipos principales: la transferencia durante el proceso de ejecución del proyecto y la transferencia del producto final.”

Durante el proceso de ejecución del proyecto la transferencia se desarrolla “cuando se involucra un actor externo desde el inicio del proyecto y las actividades del proyecto incluyen actividades de transferencia como talleres, reuniones de socialización o productos de socialización como cartillas, entre otros.” (ACEVEDO 2005).

Según Guerra y González “para transferir los frutos de una investigación científica a un proyecto determinado se deben diseñar estrategias que permitan vincular ese conocimiento adquirido por el equipo de investigadores y llevarlo a cada uno de los integrantes del proyecto, siempre buscando alternativas para que el conocimiento sea asimilado de la manera más simple posible.” (GUERRA and GONZÁLEZ 2007).

1.6.1 Factoría de Software.

Es importante señalar que el concepto de Factoría de Software surge a finales de los años 60`s e inicios de los 70`s. Estas se utilizan para definir cualquier tipo de industria en la que se obtenga un producto a través de la estandarización de su proceso de desarrollo. A continuación se enuncian varios conceptos de Factoría de Software dado por distintos autores.

Se denomina factoría, de forma genérica, a cualquier tipo de fábrica o industria, es decir, a cualquier tipo de instalación en la cual se produce la transformación de materias primas o productos semiterminados en otros productos, bien para otras industrias, bien para su uso o consumo final. Por extensión se está aplicando esta palabra para designar determinadas actividades en las cuales no se produce consumo y transformación de materias y que tienen como objeto final la obtención de productos intangibles: factoría de comunicación, factoría de cine, factoría de software. (GARCÍA, ZAYLÍ NODA and LANDEIRO 2007).

Cusumano indicó que una Factoría de Software es “una empresa productora de software que no responda a características como: producción de software en gran escala, estandarización de tareas, estandarización del control, división del trabajo, mecanización y automatización, no puede ser considerada una factoría de software. El desarrollo de una factoría implica que las buenas prácticas de Ingeniería de Software sean aplicadas sistemáticamente.” (NEGRÍN 2007).

Según Li. M “debe poseer un conjunto de herramientas estandarizadas para la construcción de software, bases históricas para ser usadas en la dirección de proyectos y principalmente poseer un alto grado de reutilización de código en el proceso de desarrollo de un determinado software, apoyado en una base de componentes reutilizables.” (NEGRÍN 2007).

En resumen una Factoría de Software es una herramienta estandarizada para la producción de software, un proceso definido con un alto nivel de reutilización de código que controla el estado de ejecución de todas las demandas.

1.6.1.1 Transferencia del Modelo de Factoría de Software.

Guerra y González señalaron que “el enfoque de Factoría de Software necesita de técnicas y estrategias para transferir el conocimiento adquirido en las investigaciones. Para llevar un Modelo de Factoría a la práctica, se definen sus entidades en paquetes, estos representan la unión de varias actividades del modelo industrial, en función del proceso que se desarrolla, las herramientas y el material de apoyo. Los paquetes son analizados y posteriormente replicados a un proceso en desarrollo. En la figura se explica el proceso de transferencia de un modelo para replicar una Factoría de Software.” (GUERRA and GONZÁLEZ 2007).

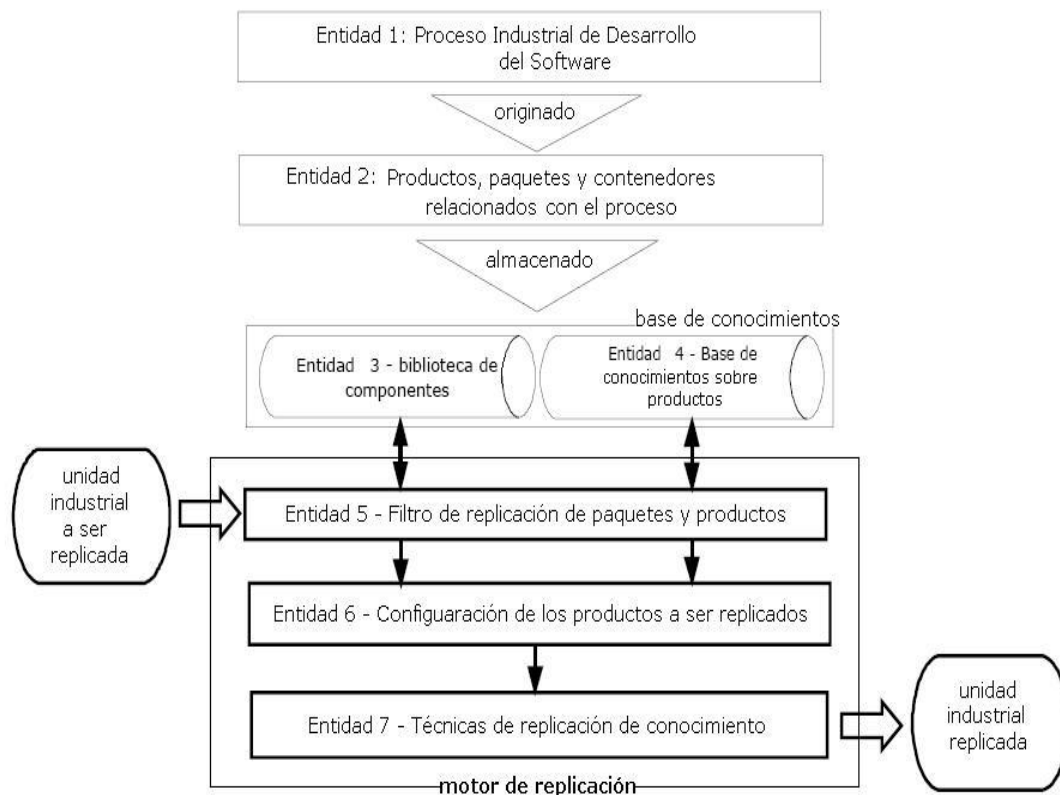


Fig. 8 Proceso de Transferencia de un Modelo para aplicar en una Factoría.

José Augusto Fabri, especialista de la universidad de Sao Paulo explica esta figura de la siguiente forma: “El proceso industrial de desarrollo del software (Entidad 1) origina los productos, los paquetes y contenedores que se replicarán (Entidad 2). Los productos deben ser guardados en una base de conocimientos que se divide en la biblioteca de componentes (Entidad 3) y la base de conocimientos sobre los productos (Entidad 4). En la biblioteca de componentes, se guardan los componentes del código e infraestructura.

En la entidad 4 el conocimiento se guarda en el proceso, herramientas y material de apoyo. Se formaliza en este trabajo el proceso industrial, los productos, los paquetes, contenedores, y las técnicas de representación del conocimiento, simplemente quedan por definir las entidades inherentes al motor de replicación.

El motor de replicación posee en su estructura las entidades 5, 6 y 7. La entidad 5 tiene como objetivo filtrar los productos, según las necesidades de la unidad a ser replicada. Hecho el filtrado, la entidad 6 configura los productos que se replicarán y la entidad 7 replica las entidades.” (JOSÉ AUGUSTO FABRI 2004)

La Transferencia del Modelo de Factoría de Software seleccionado debe estar organizada en paquetes que contengan los procesos de la misma, agrupados de acuerdo a sus funcionalidades. Una vez definidos y analizados los paquetes que serán transferidos se configuran para su implantación en la factoría. (GUERRA and GONZÁLEZ 2007).

1.7 Conclusiones.

En este capítulo se han definido los conceptos fundamentales para el desarrollo de la investigación científica de este trabajo tales como el proceso de desarrollo de software así como su mejora, la Estrategia de Transferencia y la Transferencia del Modelo de Factoría de Software. Posterior al análisis realizado sobre algunos modelos para el mejoramiento de procesos de desarrollo de software, descritas sus características, ventajas y desventajas se decide escoger el Modelo Ideal porque es el que más se adecúa a lo que se necesita para mejorar el Proceso de Transferencia del Modelo de Factoría de Software en los proyectos de SWE en la UCI. Se evidencia la importancia del Modelo Ideal pues proporciona un conjunto de actividades coherentes para sustentar la adopción de las prácticas recomendadas por el CMM; además de ser un modelo de mejora organizacional que sirve como mapa para iniciar, planificar e implementar acciones tendientes a mejorar los procesos.

Capítulo 2: Propuesta de Solución

2

2.1 Introducción.

En el presente capítulo se detallará la propuesta de solución en vista de la ya existente para el Proceso de Transferencia del Modelo de Factoría de Software, siendo esta el objetivo principal del presente trabajo. Se describen las características generales del Modelo Ideal y sus fases correspondientes, para facilitar de esta forma un mejor entendimiento del mismo. Se muestra la situación actual del Proceso de Transferencia del Modelo de Factoría de Software en la UCI. Finalmente se detalla la forma de aplicación de las fases del Modelo Ideal a dicha transferencia, para lograr un proceso mejorado y con calidad que pueda ser aplicado de forma eficiente a cualquier proyecto de SWE en la UCI.

2.2 Características generales del Modelo Ideal.

Actualmente las organizaciones están reconociendo la necesidad de aplicar métodos para guiar la adopción de nuevas herramientas y procesos. Muchos esfuerzos de la mejora, incluyendo la mejora de proceso del software, son complejos y sus efectos tienen gran envergadura, por lo que requiere un acercamiento sistemático para manejar el ciclo vital una vez implantada la tecnología. Para ayudar a satisfacer esta necesidad el SEI ha desarrollado y refinado el modelo IDEAL. Originalmente era un modelo del ciclo vital para la mejora de proceso del software basado sobre la madurez del Modelo CMM de la capacidad para el software, debido a esto el modelo utilizó términos de la mejora de proceso. Luego de ser revisado por el SEI la nueva versión del modelo se acentúa hacia la meta de una aplicabilidad más amplia.

IDEAL provee un acercamiento usable y claro a la mejora continua contorneando los pasos necesarios para establecer un programa de mejora acertado; proporcionando una aproximación disciplinada de la ingeniería. En las organizaciones que lo han implementado han logrado beneficios en muchas áreas de la mejora. El modelo es llamado de esta manera, por las diferentes fases de trabajo que involucra: **Initiating**, **Diagnosis**, **Establishing**, **Acting** y **Leverageg**. Estas poseen diferentes actividades o pasos por desarrollar. A continuación se da una breve descripción de cada una de ellas.

2.2.1 Fase de Inicialización.

El propósito de esta fase es establecer los fundamentos básicos para garantizar y dar soporte a la iniciativa de mejora de procesos. Tiene como entrada, el estímulo al cambio para el mejoramiento por parte de la organización dueña del proceso, ya sea por un desastre en sus negocios o porque deseen mejorar su competitividad a través de la calidad y la productividad de sus procesos de software. Identificada la necesidad de mejora, la alta dirección es la encargada de fijar el contexto para el trabajo que será realizado, determinando las áreas de la organización en que serán concentrados los esfuerzos y los de la estrategia de negocio se quieren llegar a mejorar. El apoyo de la alta dirección y de los gerentes en general es fundamental para el éxito del programa de mejoramiento. Se debe definir la disponibilidad de recursos, la infraestructura y la priorización del proyecto de mejoramiento. Las actividades de la fase de inicialización son críticas, si se hacen totalmente y bien, las actividades subsecuentes pueden proceder con la interrupción mínima. Si se hacen mal, incompletos o casuales, después el tiempo, el esfuerzo, y los recursos serán perdidos en fases subsecuentes.



Fig. 9 Fase de Inicialización.

2.2.2 Fase de Diagnóstico.

Durante esta fase se deben desarrollar dos caracterizaciones de la organización, el estado actual y el estado deseado. Estos estados de la organización se utilizan para desarrollar recomendaciones que sugieran una manera de proceder en las actividades. Generalmente esta fase es desarrollada con el asesoramiento de expertos en el modelo de referencia.



Fig. 10 Fase de Diagnóstico.

2.2.3 Fase de Establecimiento.

El propósito de esta fase es realizar la planificación específica de las mejoras que se desean alcanzar. En ella se eligen prioridades para la acción, necesidades urgentes, efectividad de la acción, impacto y otros aspectos que fueron sugeridos durante la fase de diagnóstico, teniendo en cuenta las operaciones a mejorar y las condiciones más apremiantes de cambio que tenga la organización. Luego se desarrolla una primera aproximación estratégica de trabajo en donde ya se especifica la disponibilidad de los recursos a utilizar, los cuales fueron tenidos en cuenta en la fase de inicialización. Se deben establecer los factores técnicos necesarios de ser instalados, las nuevas habilidades y conocimientos requeridos para usar una tecnología así como los factores no técnicos, incluyendo la cultura de la organización, fuentes probables de la resistencia y niveles de patrocinio. Al finalizar esta fase se debe contar con un plan de acción que incluya tareas, hitos, puntos de decisión, recursos, responsabilidades, mecanismos de seguimiento y riesgos con sus respectivas estrategias de mitigación.



Fig. 11 Fase de Establecimiento.

2.2.4 Fase de Ejecución o Actuación.

Esta fase comienza con el desarrollo de una solución que implica la definición e integración de los procesos, herramientas, información, conocimiento y habilidades, tanto existentes como nuevas. Una vez que las soluciones han sido diseñadas, se necesita probarlas en proyectos pilotos antes de decidir institucionalizarlas en el resto de los proyectos. Puede suceder que se realice más de un ciclo de prueba de la posible mejor solución, dado que se deben dar tantas pruebas como sean necesarias para que esta solución se convierta en una solución satisfactoria para el equipo de trabajo. Después de tener una solución aceptable, se procede a aplicarla a lo largo de la organización.

Es significativo tener en cuenta que cada una de las actividades desarrolladas durante la fase de ejecución debe dársele seguimiento por parte de la gerencia de la organización en conjunto con el equipo que este llevando a cabo el mejoramiento, con el objetivo de mantener controlado el plan realizado y verificando que las condiciones identificadas para cada una de las fases se estén presentando tal como se espera.



Fig. 12 Fase de Ejecución.

2.2.5 Fase de Aprendizaje.

El propósito de esta fase es aprender de la experiencia del ciclo realizado y aumentar la habilidad de la organización para mejorar los procesos en forma continua. Se identifica el esfuerzo al lograr los propósitos deseados, se recolectan, analizan, resumen y documentan las lecciones. Además se vuelven a examinar las necesidades de la empresa identificadas en la fase de inicialización para ver si fueron satisfechas. Finalmente se plantean y documentan recomendaciones que resultan del análisis y la validación; se proponen pautas y acciones para el siguiente plan de mejora.



Fig. 13 Fase de Aprendizaje.

2.3 Proceso de Transferencia del Modelo de Factoría de Software en la UCI.

En la UCI, en el proceso productivo de SWE a medida que se implementa una Estrategia de Transferencia del Modelo de Factoría de Software se requiere que los integrantes del proyecto se centren seriamente en esta misión. En la puesta en práctica del Proceso de Transferencia del Modelo de Factoría de Software en uno de los proyectos de la Universidad se ha podido apreciar que se cubre una gama de actividades directivas y una secuencia de procesos de evaluación que controlan el flujo de trabajo de la organización. En la Universidad esta transferencia se ha aplicado a varios proyectos entre los que se encuentran CNTI-Contenidos Educativos de la Facultad 8 y en el Polo de Gestión de la Facultad 3 en el proyecto Informatización del Convenio Cuba-Venezuela, dicha transferencia requiere de un modelo que sirva como guía global de las actividades por realizar para el mejoramiento de este proceso. Son muchos los modelos desarrollados a nivel internacional para la mejora de este tipo de procesos, donde luego de un detallado estudio de los mismos se selecciona el Modelo Ideal para la aplicación de forma eficiente de este tipo de transferencias.

2.4 Propuesta para la Mejora del Proceso de Transferencia del Modelo de Factoría de Software en los Proyectos de SWE.

En la facultad 8, en el proyecto CNTI- Contenidos Educativos se aplicó la Estrategia de Transferencia del Modelo de Factoría de Software en el cual no se tuvo mucho éxito debido a varios factores que influyeron en su ejecución; por esta razón se aplicará el Modelo Ideal para mejorar el proceso de transferencia y de esta manera poder aplicarlo de forma satisfactoria a cualquier proyecto de SWE en la UCI.

2.4.1 Aplicación de la Fase de Inicialización al Proceso de Transferencia.

En esta fase es donde se identifica el objetivo general para tenerlo en cuenta dentro del plan de mejoramiento; en este caso será la necesidad de mejora del Proceso de Transferencia del Modelo de Factoría de Software. Una vez que se tiene identificada la necesidad de mejora, la gerencia de la organización es la encargada de fijar el contexto para el trabajo que será realizado, identificando de esta manera en las áreas de la organización en que serán concentrados los esfuerzos y los puntos de la estrategia de negocio que se quieren llegar a mejorar.

Entre las áreas a mejorar su calidad se encuentran la Gestión de Proyecto, el Personal, las Bases Tecnológicas y el Centro de Inteligencia. En la Gestión de Proyecto, específicamente en la Gestión de la Calidad se detallarán los procesos definidos en la misma. Además, se definirán nuevos roles que se consideren necesarios para el desarrollo del proyecto así como nuevos instrumentos propuestos por el Proceso de Software Personal (PSP), todo esto en el área referente al Personal. En cuanto a las Bases Tecnológicas se pretende modernizar las herramientas existentes, además de adicionar otras. Por su parte dentro del Centro de Inteligencia se tiene como punto clave la capacitación del personal mediante la elaboración de cursos útiles para los integrantes del proyecto facilitando de esta forma el trabajo de los mismos. Finalmente se debe definir la infraestructura con la cual se llevará a cabo el trabajo, incluyendo los grupos de personas con que se contarán, es decir, las personas involucradas dentro del proyecto con el plan de mejoramiento, por consiguiente se debe crear un Grupo de Proceso de Ingeniería de Software (SEPG) que será el núcleo de la mejora del proceso. También se debe contar con un Grupo de Gestión de Manejo (MSG) para asesorar a la SEPG y supervisar sus actividades. Además se debe crear un Grupo de Trabajo Técnico (TWG), que será el líder del área que esté en proceso de mejora.

Otro elemento a tener en cuenta lo constituye la capacitación del personal del proyecto sobre el modelo de mejora que se está aplicando, así como del modelo CMM ya que el Modelo Ideal utiliza las prácticas recomendadas planteadas por este. Para dar cumplimiento a lo antes expuesto se decide la elaboración de un curso de capacitación para que de esta manera los implicados en la mejora adquieran los conocimientos necesarios sobre el modelo que se va a aplicar.

Seguidamente se expone como quedará estructurado el curso:

Curso de Mejora de Proceso (Modelo Ideal).

- Procesos y conceptos relacionados.
- Visión general de la Mejora de Procesos de Software.
 - Definición de la Mejora de Procesos.
 - Ventajas de llevar a cabo un programa SPI.
- Modelos de Mejora de Procesos más conocidos (CMM, Bootstrap, ISO / IEC 15400, Mesopayme, IMPACT).

- Características de cada modelo.
- Actividades o Fases.
- Ventajas y Desventajas.
- Modelo Ideal.
 - Características generales.
 - Fases:
 - Inicialización.
 - Diagnóstico.
 - Establecimiento.
 - Ejecución.
 - Aprendizaje.
 - Aplicación del modelo.
 - Importancia de aplicar el Modelo.

Una vez impartido el curso los estudiantes del equipo de desarrollo tendrán los siguientes beneficios:

- Obtendrán la información básica necesaria para decidir cuándo y por dónde empezar un programa de SPI.
- Visión general de los principales modelos para evaluación de procesos.
- Mecánica de los programas de mejora: cómo iniciarlos, conducirlos e implantarlos.
- Información detallada del Modelo Ideal lo que facilitará la aplicación del mismo en los proyectos.

2.4.2 Aplicación de la Fase de Diagnóstico al Proceso de Transferencia.

A continuación se expondrá la puesta en práctica de la fase de diagnóstico, en la cual se caracterizará la situación actual del Proceso de Transferencia del Modelo de Factoría de Software y los estados en los cuales se quiere que se encuentre en un futuro este proceso en los proyectos de SWE, además de las recomendaciones del trabajo que se consideren necesarias.

Según Guerra y González la estrategia para hacer la transferencia del Modelo de Factoría de Software aplicando inteligencia es la siguiente:

1. Realizar un proceso de investigación para recopilar la información que permita implantar un Modelo de Factoría a un proyecto.
2. Definir el proceso de transferencia para el proyecto.
3. Gestionar todos los recursos necesarios para poner en marcha la estrategia de transferencia.
4. Implantar el proceso de transferencia definido al proyecto.
5. Hacer un seguimiento del funcionamiento del proceso de transferencia.
6. Emitir un reporte de los resultados obtenidos. (GUERRA and GONZÁLEZ 2007).

En el transcurso de la aplicación del Modelo Ideal a este proceso se desea que se cumpla la siguiente expectativa:

Obtener un proceso de desarrollo de software, en este caso la Estrategia de Transferencia del Modelo de Factoría de Software mejorado, de forma tal que sea posible aplicarlo con éxito a cualquier proyecto de SWE de la UCI.

A modo de Recomendaciones se tiene que:

- Lograr que los involucrados en el proyecto tengan un alto conocimiento sobre el proceso que se está llevando a cabo y un compromiso constante con este.
- Obtener una mejor definición de los roles para la Factoría de Software en los proyectos de SWE.
- Conseguir que cada uno de los integrantes del equipo de trabajo apliquen todos los instrumentos del PSP para de esta forma se realice una correcta planificación y revisión del trabajo.
- Alcanzar el grupo de herramientas adecuado para utilizar en cualquier proyecto de SWE.

2.4.3 Aplicación de la Fase de Establecimiento al Proceso de Transferencia.

En este acápite se muestra la implantación de la fase de establecimiento, delimitando sus objetivos y estos a su vez por los niveles correspondientes de forma tal que se logre un mejor entendimiento de los mismos; una vez que estos objetivos están organizados de la forma antes expuesta se priorizan para una mejor realización de estos. Para una mejora del proceso se

procede a la realización de un Plan de Acción compuesto por actividades, hitos, responsables, tiempo de duración, riesgos y estrategias de mitigación.

A continuación se plantea la organización de los objetivos durante el ciclo de trabajo establecido, quedando organizados en 3 niveles donde los objetivos de cada nivel son complementarios y se comenzará a trabajar con los de los niveles de mayor prioridad de especificidad.

Nivel 1: Objetivos a largo plazo.

1. Mejorar el Proceso de Transferencia del Modelo de Factoría de Software.

Nivel 2: Objetivos a mediano plazo.

1. Mejorar la entidad Gestión de Proyecto.
2. Mejorar la entidad Persona.
3. Mejorar la entidad Bases Tecnológicas.
4. Mejorar la entidad Centro de Inteligencia.

Nivel 3: Objetivos inmediatos.

1. Perfeccionar la Gestión de la Calidad.
2. Reformar los roles de los integrantes del proyecto.
3. Adicionar nuevos Instrumentos del PSP a utilizar.
4. Modernizar y adicionar nuevas herramientas.
5. Reestructurar la capacitación del personal que integra el proyecto.

Priorización de los objetivos.

Una vez que se encuentran estructuralmente delimitados los objetivos y luego de analizar cuales serían los de mayor necesidad de cumplimiento se decide que sean los objetivos correspondientes al Nivel 3, debido a que son los más específicos y detallan con mayor exactitud lo primordial a realizar; luego de haber desarrollado el objetivo 1 correspondiente a este nivel se estará dando cumplimiento al objetivo 1 del Nivel 2, lo mismo ocurre en el caso de los objetivos 2 y 3 del Nivel 3 ya que se habrá cumplido el 2 del Nivel 2. Igualmente realizando el 4 y 5 del Nivel 3 se habrá implementado el objetivo 3 y 4 del Nivel 2 respectivamente.

Finalmente cuando se encuentran ejecutados o desarrollados todos los objetivos del Nivel 2 se logra efectuar el objetivo propio del Nivel 1, el cual sería el objetivo general de esta investigación, ya que es el que se le debe dar una rectificación eficiente para su óptima puesta en práctica en la producción de SWE en la UCI.

Tabla 2 Plan de Acción.

Actividades	Hitos	Respon- sables	Tiempo de duración	Riesgos	Estrategia de Mitigación
Mejorar la entidad Gestión de Proyecto.	Documentación de la Gestión de la Calidad.	Grupo de Trabajo Técnico.	8 días	1. Inexperiencia de los integrantes del proyecto de los estándares de calidad a utilizar.	1.1 Organizar reuniones con los expertos de calidad a nivel central y los integrantes del proyecto sobre las dudas que se tengan sobre los estándares.
Mejorar la entidad Persona.	Roles e Instrumentos PSP.	Grupo de Trabajo Técnico.	7 días	1. No correspondencia de los roles con el número de personas involucradas.	1.1 Si la cantidad de roles es mayor que la de personas, una persona puede desarrollar más de un rol. 1.2 Si la cantidad de

				<p>2. Desconocimiento por parte de los integrantes del proyecto de los instrumentos del PSP a utilizar.</p>	<p>personas es mayor que la cantidad de roles, un rol puede ser desempeñado por varias personas. 2.1 Realizar talleres sobre como trabajar con cada uno de los Instrumentos del PSP.</p>
<p>Mejorar la entidad Bases Tecnológicas</p>	<p>Nuevas Herramientas</p>	<p>Grupo de Trabajo Técnico.</p>	<p>8 días</p>	<p>1. Inexperiencia por parte de los integrantes del proyecto del funcionamiento de las herramientas propuestas.</p>	<p>1.1 Impartir cursos a todos los integrantes sobre el funcionamiento de las herramientas que se proponen utilizar.</p>
<p>Mejorar la entidad Centro de Inteligencia.</p>	<p>Medios de capacitación para los integrantes del</p>	<p>Grupo de Trabajo Técnico.</p>	<p>7 días</p>	<p>1. Mal aplicación de los cursos de capacitación del personal.</p>	<p>1.1 Verificar la manera de aplicar los cursos con personas</p>

	proyecto.				expertas en la materia.
--	-----------	--	--	--	-------------------------

2.4.4 Aplicación de la Fase de Ejecución al Proceso de Transferencia.

En esta fase se comienza con el desarrollo de la solución. Se pretenden realizar cambios en algunas áreas de mejora, como es el caso de la Gestión de Proyectos, el Personal, las Bases Tecnológicas y el Centro de Inteligencia. En cada una de estas áreas existen puntos claves que se quieren llegar a mejorar, para finalmente lograr el objetivo principal del presente trabajo definido en la fase de inicialización.

2.4.4.1 Gestión de Proyecto.

Un punto fundamental en el desarrollo de cualquier tipo de proyecto lo constituye la Gestión de Proyecto, la cual tiene como propósito asegurar que los proyectos contribuyan al cumplimiento de los objetivos de la organización. Dentro de esta, uno de los eslabones principales es la Gestión de la Calidad del Software.

En el Proceso de Transferencia del Modelo de Factoría de Software, Guerra y González definen tres procesos dentro de la Gestión de la Calidad; Planificación de la Calidad, Control de la Calidad y Aseguramiento de la Calidad (SQA).

En cuanto al proceso de Planificación de la Calidad se propone continuar utilizando en los proyectos de SWE el estándar ISO 9001 para determinar la calidad de los productos y CMMI para la mejora de procesos de los proyectos.

Por otra parte, dentro del proceso de Control de la Calidad se debe tener en cuenta que el objetivo primordial es la reducción de los riesgos. Por cada objetivo de control/riesgo potencial se deben identificar las técnicas de control existentes que deben minimizar el riesgo.

Para llevar a cabo un buen Control de la Calidad en los proyectos se deben realizar evaluaciones, inspecciones, auditorías y revisiones que aseguren que se cumplan las responsabilidades asignadas, se utilicen eficientemente los recursos y se logre el cumplimiento de los objetivos del producto.

Además el Control de Calidad debe incluir la verificación del proceso de desarrollo de software para garantizar que los estándares seleccionados o escogidos para el proceso se están cumpliendo de la forma correcta.

El Aseguramiento de Calidad es otro de los procesos referentes a la Gestión de la Calidad. Entre sus principales objetivos están la confiabilidad, el desempeño, la funcionalidad y la reutilización.

Las actividades del SQA del software contemplan aquellas tareas del proceso de desarrollo de software que buscan asegurar el diseño, desarrollo y distribución de una aplicación exitosa u otra forma de tecnología de software. Sus funciones están dirigidas a:

- Identificar las posibles desviaciones en los estándares aplicados, así como en los requisitos y procedimientos especificados.
- Probar que se han cumplido las medidas preventivas o correctivas necesarias.
- Realizar revisiones al software ya que son una de las actividades más importantes del SQA, debido a que permiten eliminar defectos lo más pronto posible, cuando son menos costosos de corregir.

Los principales elementos del SQA son los siguientes:



Fig. 14 Estructura del SQA.

Muchas de las actividades referentes al Aseguramiento de la Calidad a realizar deben estar incorporadas al proceso de desarrollo utilizado por el proyecto; primeramente el proyecto deberá tener definido su proceso de desarrollo de software. Teniendo en cuenta que si existen herramientas automatizadas que den soporte al SQA para probar aplicaciones y componentes de estas, registrando y ejecutando casos de prueba, así como generándolos en forma automática se lograría conseguir una productividad alta y con calidad.

2.4.4.2 El Personal.

Un aspecto fundamental en la elaboración de un software lo constituyen sin lugar a dudas los roles que compondrán el equipo de desarrollo. La cantidad de roles que se utilizan para el desarrollo de un proyecto de SWE depende en gran medida de la magnitud del proyecto. Mientras más grande y complejo sea el proyecto se necesitará de una mayor cantidad de participantes para su elaboración y más roles especializados. Otro factor importante a considerar para elegir los roles es el tiempo asignado al desarrollo del proyecto.

En la UCI para aplicar la Estrategia de Transferencia del Modelo de Factoría de Software, Guerra y González definen los siguientes roles:

- Jefe de Línea de Desarrollo.
- Gerente de la Factoría.
- Asesores de la Factoría.
- Gerente de Soporte.
- Planificador.
- Administrador del Repositorio.
- Gerente de Desarrollo.
- Revisores.
- Arquitectos.
- Programadores de Interfaz Gráfica.
- Programadores de actividades interactivas (Flasheros).
- Jefe de Prueba.

- Probadores.
- Documentadores.
- Gestor de Información Interna.
- Gestor de Información Externa.
- Gestor Empresarial.

Una vez estudiados todos los roles y responsabilidades definidos por Guerra y González para implantar la Estrategia de Transferencia del Modelo de Factoría de Software en el proyecto CNTI- Contenidos Educativos se llega a la conclusión que para aplicar esta transferencia a cualquier SWE en la UCI se deben hacer varias transformaciones entre las que se encuentran, el adiconado de algunos roles, debido a que existen actividades que no están asignadas a ningún rol de los ya establecidos, por lo que puede que no se realicen y además asignar mayor responsabilidades a algunos integrantes del proyecto. Todos estos cambios son necesarios para lograr un proceso de desarrollo de software eficiente, un trabajo por parte de los integrantes más satisfactorio y de esta forma un producto con mayor calidad.

Después de un análisis de los roles previamente definidos para la implementación de este modelo en el proyecto CNTI-Contenidos Educativos donde estos no daban una solución óptima al mismo se realizan algunos cambios. Seguidamente se expondrá como quedarían los roles para la Factoría de Software y las responsabilidades de cada uno.

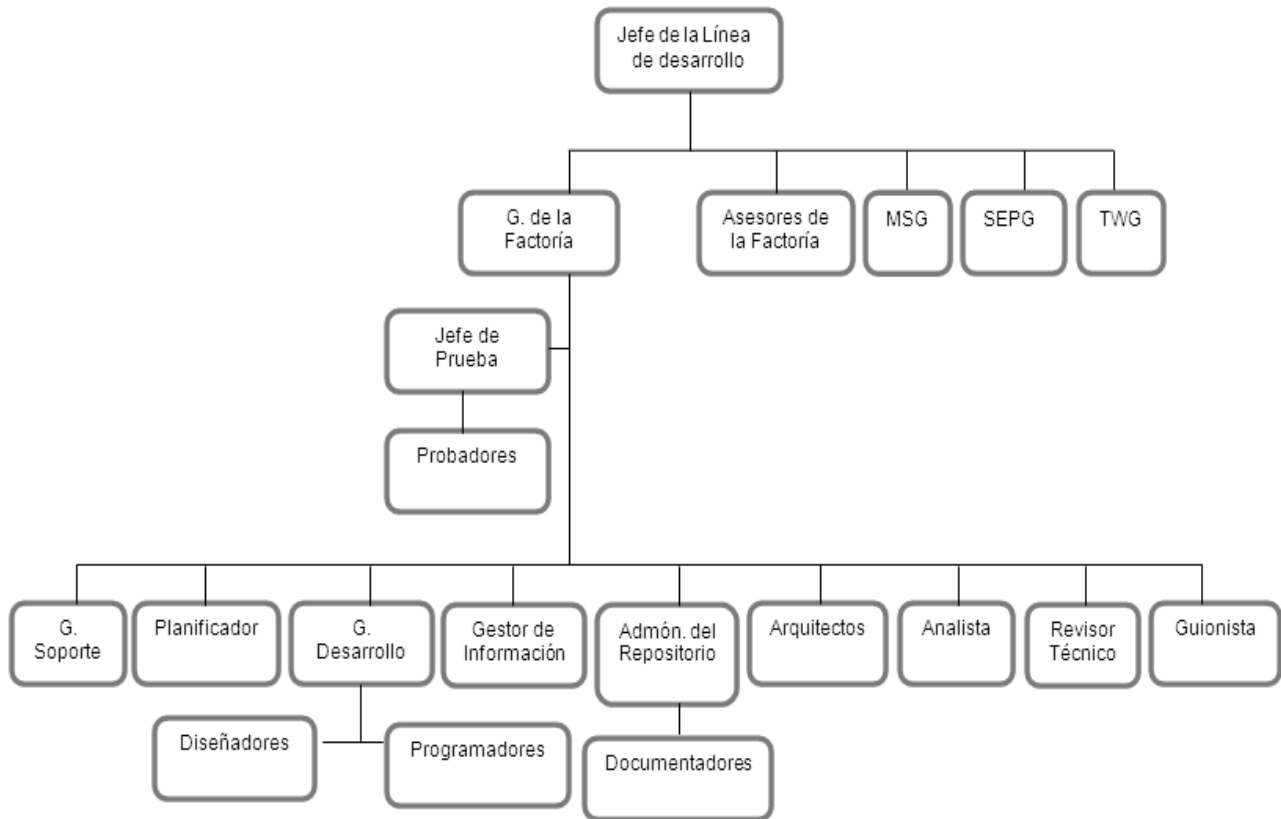


Fig. 15 Estructura Organizacional.

Jefe de Línea de desarrollo: Es el encargado de dirigir el desarrollo de productos software en una línea determinada dentro de la factoría. Tiene la misión de coordinar las reuniones de presentación entre el cliente y el gerente de la factoría, supervisar su trabajo y garantizarle los recursos para responder a la solicitud del cliente.

Gerente de la Factoría: Participa en la definición del proyecto y la metodología de desarrollo. Es el encargado de reunirse con los clientes, gestiona los recursos y materiales necesarios para el proyecto y para el equipo de desarrollo. Aprueba las tecnologías a usar en el desarrollo del proyecto. Responsable de coordinar y organizar las tareas que se asignan a los miembros del equipo de desarrollo.

Asesores de la Factoría: Tienen la responsabilidad de orientar al gerente de la factoría en la implantación y hacerle un estudio y seguimiento a la misma para poder evaluar los resultados que se obtengan.

Gerente de Soporte: Es la persona encargada de instalar las herramientas y verificar el funcionamiento de los puestos de trabajo, controlando tanto el hardware como el software.

Planificador: Planifica las reuniones con los trabajadores de la factoría. Mantiene actualizado el plan de proyecto, cronogramas y fechas de entregas. Lleva el control de la ejecución del proyecto emitiendo informes periódicos del estado de avance del mismo. Mide la eficiencia del desarrollo, establece métricas, controla los tiempos de ejecución, imprevistos, contratiempos.

Administrador del Repositorio: Configura el repositorio de componentes y establece las políticas de seguridad. Actualiza la información de cada componente y gestiona las solicitudes de almacenamiento y reutilización.

Gerente de Desarrollo: Para desarrollos grandes, el software se divide en subsistemas que se construye en equipos dirigidos por un jefe de sistema o de desarrollo. Guía a cada equipo de desarrollo y asigna las tareas a cada uno de los integrantes de su equipo. Verifica el cumplimiento del cronograma de trabajo, las plantillas elaboradas y controla la calidad de los procesos de desarrollo.

Revisor Técnico: Revisa los guiones de contenidos y las diagramaciones para cada producto de SWE para corregir los errores encontrados.

Arquitecto: Se encarga de la definición de la arquitectura que guiará el desarrollo y de la continua refinación de la misma en cada iteración; debe construir cualquier prototipo necesario para probar aspectos riesgosos desde el punto de vista técnico del proyecto; definirá los lineamientos generales del diseño y la implementación.

Programadores de Interfaz Gráfica: Son los encargados de montar las medias a cada página Web, prueban su funcionalidad y solicitan a su gerente de desarrollo una revisión.

Programadores de actividades interactivas (Flasheros): Programan las actividades interactivas como son los juegos y preguntas de selección. Prueban la funcionalidad de los mismos y después solicitan una revisión.

Jefe de Prueba: Planifica las pruebas, diseña los procedimientos y casos de prueba para evaluar los resultados obtenidos. Elabora un reporte de los errores encontrados y se los envía al gerente de la factoría para corregirlos y pasar a otra fase del producto.

Probadores: Prueban los procedimientos y casos de prueba diseñados por el jefe de prueba. Elaboran un listado de errores.

Documentadores: Documentan los componentes y elaboran una plantilla de información de los componentes.

Gestor de Información: Gestiona los recursos técnicos y humanos para los cursos de capacitación. Almacena y actualiza los documentos de información. Investiga el mundo exterior a la factoría en cuanto a la tecnología y promueve la investigación en función de las necesidades y demandas del cliente. Este cargo lo pueden realizar varias personas en dependencia de la carga de trabajo que exista en este ámbito.

Diseñadores: Interpretan la información resultante del análisis y lo traducen al lenguaje en el cual se va a programar el producto (interfaz, negocio y acceso a datos). Estos son los encargados de definir además los elementos de diseño a tener en cuenta para la implementación de los casos de uso. Al igual que diseñan la implementación sobre la arquitectura definida. Integran los componentes de la solución y definen las interfaces, dirigiendo a su vez el trabajo de los programadores.

Analistas: Definen una estrategia para la captura de requisitos. Precisan los artefactos que se obtendrán como resultado del análisis y la metodología que se sigue para obtenerlos. Delimitan las técnicas de recopilación de información que serán usadas durante la captura de requisitos. Supervisan y controlan el cumplimiento de la metodología para el análisis, definiendo los sistemas, subsistemas y módulos en los que se organiza la solución del software a elaborar.

Guionista: Es el encargado de confeccionar los guiones teniendo en cuenta la solicitud de los clientes. Revisan y controlan constantemente que se cumpla lo establecido en los guiones.

Grupo de Gestión de Manejo (MSG): Es el encargado de vincular el programa de mejora con las expectativas y el objetivo de la organización. Realiza una estimación del esfuerzo y los recursos. Efectúa el seguimiento del progreso de la mejora y orienta a los involucrados sobre esta.

Grupo de Proceso de Ingeniería de Software (SEPG): Supervisa y refuerza el apoyo a SPI, sosteniendo las actividades de mejora. Identifica y coordina las actividades a mejorar, documentando la eficiencia que se va teniendo en las distintas áreas.

Grupo de Trabajo Técnico (TWG): Se centra en la mejora de un área en específico. Es el encargado de documentar y mejorar los procesos luego de identificar los problemas y las soluciones. Crea e implementa el plan de acción para la ejecución de la solución.

Luego de definir los roles de la Factoría de Software a utilizar en la Estrategia de Transferencia se exponen a continuación los instrumentos del PSP que se emplearon en el proyecto CNTI-Contenidos Educativos de la facultad 8 y como quedarán para cualquier proyecto de SWE en la UCI que implante este tipo de estrategia.

Según Humphrey el “Proceso Software Personal (PSP) fue diseñado para ayudar a los ingenieros del software a hacer bien su trabajo. Muestra la manera de aplicar métodos avanzados de ingeniería a sus tareas diarias. Proporciona métodos detallados de planificación y estimación, muestra a los ingenieros a controlar su rendimiento frente a estos planes y explica la forma en que los procesos definidos guían su trabajo.” (HUMPHREY 2001).

En la Estrategia de Transferencia del Modelo de Factoría de Software implantada por Guerra y González en la entidad Personas del Modelo de Factoría aplicando Inteligencia se propone que los trabajadores implicados en la elaboración del producto utilicen los siguientes instrumentos del PSP.

- Cuaderno del ingeniero: Permite registrar tiempos, guardar cálculos, tomar notas de las fases de desarrollo y registrar además cualquier otro tipo de información que afecte su horario laboral. (Ver Anexo 1).
- Cuaderno de registro de tiempo: Sirve para registrar diariamente las actividades realizadas en cada instante del día. Cuaderno muy útil para contabilizar y describir las interrupciones ocurridas durante la jornada laboral, y dejar almacenado el historial de las mismas, ayudándolos de ese modo a mejorar la calidad y eficiencia de su trabajo. (Ver Anexo 2).
- Resumen semanal de actividades: Se agrupan las actividades por días, obteniendo totales de tiempo relacionados con los trabajos efectuados diariamente y en la semana. Es muy útil para que los directivos de la factoría planifiquen las actividades. (Ver Anexo 3).
- Cuaderno de registro de defectos: Permite reunir los datos de defectos. Registra la información de los defectos para tenerlos presente en futuras revisiones. Permite llevar un

control detallado de los defectos encontrados, cuándo se eliminó y el tiempo que demoró la corrección. (Ver Anexo 4). (GUERRA and GONZÁLEZ 2007).

Teniendo en cuenta lo que representa el PSP para los ingenieros de software se llega a la consideración que para una mejor aplicación de este se deben utilizar todos sus instrumentos porque cada uno cumple una función específica y en su conjunto son una guía de prácticas personales que ayudan al equipo de desarrollo a planificar y revisar su trabajo, produciendo así software con óptimas características operacionales y una alta calidad. Luego de una eficaz puesta en práctica de los instrumentos del PSP se podrán obtener datos que expongan la efectividad del trabajo así como lo más significativo con respecto a las debilidades y puntos fuertes de cada individuo. Por todo lo antes expuesto se adiciona a la anterior propuesta los siguientes instrumentos en pos de lograr una mayor satisfacción de la aplicación de la estrategia:

- Cuaderno de Trabajos: Se utiliza para registrar los datos de tiempos estimados y reales, además de ser un documento de planificación del producto, pues trata datos del mismo. Tiene como ventaja que proporciona una forma concisa de registrar y acceder a una gran cantidad de datos históricos de proyectos; siendo la clave para hacer estimaciones exactas y estas a su vez las claves para realizar buenas planificaciones. (Ver Anexo 5).
- Lista de Compromisos: Es el documento en el cual se plasma la información correspondiente al trabajo de los ingenieros en cuanto a los compromisos, es por eso que resulta importante gestionarlos para de esta manera no olvidarlos. Los integrantes de los proyectos participan en revisiones, escriben informes, asisten a reuniones, hacen correcciones de programas y actualizan módulos de programas por lo que se hace necesario este instrumento para no dejar pasar los compromisos. También son de gran ayuda cuando el trabajo que se necesita hacer exceda el tiempo disponible. (Ver Anexo 6).
- Plan del Proyecto: Proporciona un marco de trabajo para gestionar la revisión y el control de los proyectos. Provee una definición de cada tarea principal, además de una estimación de los recursos necesarios y del tiempo. Si se cuenta con un Plan del Proyecto documentado es útil para compararlo con el rendimiento real, permitiendo obtener los errores de la planificación y de esta forma mejorar la misma. (Ver Anexo 7).

Luego de que se agreguen a la anterior estrategia los instrumentos del PSP antes mencionados, se prevé que se tenga un mayor rendimiento en cuanto a la aplicación de la misma en los proyectos de SWE en la UCI, para garantizar de esa forma que estos puedan corresponderse con las necesidades de los usuarios finales y que los productos cuenten a su vez con la calidad requerida, permitiendo que la institución gane en prestigio.

2.4.4.3 Bases Tecnológicas.

Un aspecto de gran importancia para poder llevar a cabo una estrategia de esta índole lo constituyen las Bases Tecnológicas, en la que van a estar incluidos varios elementos como es el caso de las herramientas, las que garantizarán el soporte para una realización exitosa del proyecto en cuestión.

Actualmente en la UCI se está llevando a cabo un proceso de transición del Software Propietario al Software Libre, ya que su aporte a la economía es significativo, precisamente por ser alternativas libres para la creación de todo tipo de proyectos principalmente los de gran envergadura tanto para el país como para la exportación. Debido a que esta migración avanza lentamente para lo que realmente requiere la producción actual, los proyectos que desarrollan SWE en su mayoría emplean herramientas propietarias pues son de las que se tiene mayor conocimiento y experiencias de su uso en la producción. Por esta razón se propondrán dos versiones de las herramientas a utilizar en los proyectos de SWE, una para Software Propietario y otra para Software Libre.

En la entidad Bases Tecnológicas de la tesis “Estrategia de Transferencia del Modelo de Factoría de Software aplicando inteligencia para Contenidos Educativos”, Guerra y González plantean un grupo de herramientas a utilizar: Dreamweaver / IDE, Flash MX 2004, Microsoft Project y TortoiseSVN. Luego de una amplia investigación y considerando que la estrategia en cuestión es para aplicarse a productos de SWE se decide incorporar las siguientes herramientas, para así conformar la versión para Software Propietario:

- **CorelDRAW Graphics Suite** es una herramienta que se ha mantenido en el liderazgo de software de diseño gráfico e ilustración vectorial para computadoras personales. Además de haber sido el primer producto de diseño gráfico para Windows, CorelDRAW Graphics Suite incluye tres potentes aplicaciones: CorelDRAW para la ilustración vectorial y diseño

de páginas, Corel PHOTOPAINT para la edición profesional de imágenes digitales y Corel R.A.V.E para la creación de gráficos en movimiento.

- **Macromedia Fireworks 8** es una herramienta que puede utilizarse para crear, editar y animar gráficos Web, añadir interactividad avanzada y optimizar imágenes en entornos profesionales. Es posible crear y modificar imágenes vectoriales y de mapa de bits en una sola aplicación. Todo es modificable en todo momento. El flujo de trabajo puede automatizarse para satisfacer las necesidades de cambio y actualización que de otra forma exigirían una enorme dedicación.
- **Rational Rose** es una herramienta de modelación visual. Posee una plataforma independiente que permite crear, ver, modificar y manipular los componentes de un modelo, los cuales contienen vistas, compuestas por diagramas, artefactos y sus especificaciones, pudiéndose agrupar en paquetes. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software UML, la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. En esta herramienta se puede generar código para diferentes lenguajes como: Visual Basic, Java, Oracle, Corba.
- **MySQL** es un sistema de gestión de bases de datos, muy rápido, fiable y fácil de usar. Se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. Trabaja en entornos cliente/servidor.

Por otra parte como la era de la información avanza aceleradamente en conjunto con las tecnologías y estas a su vez con las herramientas correspondientes se hace necesario la utilización de las versiones más actualizadas de las mismas como es el caso de la herramienta Flash MX 2004 por el de Macromedia Flash 8 ya que esta versión dispone de nuevas funciones como por ejemplo las que a continuación se exponen:

- Posibilidades de creación de efectos gráficos adicionales.
- Codificador de video independiente e integrado.
- Compatibilidad con transparencias de canales alfa en video.

- Calidad de video superior.
- Representación de texto de alta calidad con control avanzado.
- Herramientas de texto mejoradas.
- Nuevo complemento de video para exportar archivos Flash Video (.FLV) de productos de video profesionales.

A continuación se expone el grupo de herramientas a utilizar que conformará la versión de Software Libre.

- **Zend Studio** es el soporte en desarrollos y pruebas de PHP más completo, con herramientas para la creación de aplicaciones altamente fiables. Asegura el desarrollo de software mediante la combinación del IDE líder para PHP con un entorno de prueba que agiliza la seguridad de la calidad, integración y las etapas de los procesos. Brinda todo lo que necesita para construir, probar y entregar aplicaciones PHP de alto rendimiento.
- **DotProject** es una aplicación web basada en PHP que incluye módulos para compañías, proyectos, tareas (con diagramas de Gantt), foros, ficheros, calendario, contactos, ayuda de escritorio, soporte multilinguaje y módulos con permiso. Corre tanto en Windows como en Linux. Con licencia GPL.
- **RapidSVN** proporciona una interfaz fácil de usar para las características de Subversión. Muy sencillo para los principiantes, pero lo suficientemente flexible como para aumentar la productividad de los usuarios experimentados de Subversión. Es una herramienta que corre en cualquier plataforma en la que Subversión y wxWidgets puedan ejecutar: Linux, Windows, Mac OS / X y Solaris, completamente escrito en C + +. Ha sido traducido a muchos idiomas entre los que se encuentra alemán, francés, italiano, portugués, ruso, chino simplificado, japonés.
- **GIMP** es un programa de edición de imágenes, libre y gratuita, englobada en el proyecto GNU y disponible bajo la licencia GPL. GIMP lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, bitmap, gif, png, pcx, tiff, y los de Photoshop, además de poseer su propio formato de almacenamiento de ficheros, xcf. También es capaz de importar ficheros en pdf e imágenes vectoriales en formato svg creadas, por ejemplo, con Inkscape. Es una herramienta que corre en Linux, Windows y Mac OS / X.

- **PostgreSQL** es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD. Contiene una documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Altamente adaptable a las necesidades del cliente. Posee soporte para los lenguajes más populares del medio entre los que se encuentran PHP, C, C++, Perl y Python. Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos y Windows.
- **Visual Paradigm** es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta CASE posibilita la generación de código para Java y exportación como HTML; con licencia gratuita y comercial.
- **MTASC** es un compilador de código abierto para ActionScript 2.0. Funciona desde la línea de comandos y por lo tanto obliga a trabajar con Programación Orientada a Objetos (POO), o sea los proyectos tienen que ser un conjunto de clases. Es independiente de plataforma, se puede compilar para la plataforma que se desee, lo único que se necesita es poder compilar el lenguaje y el compilador de Ocalm.
- **SWFTools** son un grupo de herramientas de código abierto para crear y manipular ficheros SWF. Ha sido liberado bajo licencia GPL, y funciona en entornos Windows, Mac OS X, Linux y otros sistemas tipo Unix.
- **SWFMILL** es un traductor de XML a un swf y los swf en XML, la sintaxis es fácil, sirve para agregar objetos externos al swf. Se utiliza para la generación de componentes de la librería conteniendo imágenes (PNG y JPEG), fuentes (TTF) u otras películas swf para su uso con los compiladores de ActionScript MTASC o haXe, además swfmill puede ser usado para producir estructuras swf tanto complejas como simples.

2.4.4.4 Centro de Inteligencia.

Según plantea Casañola, la entidad Centro de Inteligencia del Modelo de Factoría de Software aplicando inteligencia es la encargada de “realizar la gestión del conocimiento para saber las potencialidades y los problemas de la factoría, debe manejar la información interna, o sea, la

gestión organizacional. La vigilancia tecnológica, la inteligencia empresarial y la prospectiva es la información externa. Donde:

La gestión del conocimiento es la gestión de los activos intangibles que generan valor para la organización. La mayoría de estos intangibles tienen que ver con procesos relacionados de una u otra forma con la captación, estructuración y transmisión del conocimiento. Por lo tanto, la gestión del conocimiento tiene en el aprendizaje organizacional su principal herramienta. Con esto se pretende transferir el conocimiento y experiencia existente en los desarrolladores, de modo que pueda ser utilizado como un recurso disponible para otros en la factoría.

La vigilancia tecnológica tiene el objetivo de conseguir que cada persona de la empresa tenga toda la información disponible que necesita para la toma de decisiones. Permite conocer el presente para orientar el futuro.

La inteligencia empresarial brinda la capacidad para reunir, organizar, analizar cualitativa y cuantitativamente para obtener conocimiento que permita la toma de decisiones con menor incertidumbre y orientación estratégica a mediano y largo plazo.

La prospectiva (del inglés "prospect", significa esperanza), es la disciplina que estudia el futuro desde un punto social, científico y tecnológico con la intención de comprenderlo y de poder influir en él. Gaston Berger uno de los fundadores de la disciplina la definía como la ciencia que estudia el futuro para comprenderlo y poder influir en él." (CASAÑOLA 2007).

Teniendo en cuenta que la entidad Centro de Inteligencia es la que se dedica a la gestión de los recursos humanos de los proyectos, elemento fundamental y decisivo para el desarrollo exitoso de estos, se debe tener en cuenta que en los mismos exista un rol que se encargue de este aspecto, que en este caso será el Gestor de Información el cual gestionará la información interna y externa de un determinado proyecto. Este rol lo pueden ejecutar una o varias personas en dependencia de la carga de trabajo que exista en el proceso de desarrollo en cuestión. Analizará el mercado exterior para valorar los costos de todos los recursos de la factoría, así como la de los productos similares a los propuestos a elaborar o en proceso de elaboración. Además de promover el conocimiento a todos los miembros del equipo de desarrollo. También tiene la responsabilidad de almacenar y mantener actualizados los documentos de información generados en la factoría y los que están relacionados con el

conocimiento que se maneja y la de buscar las vías para la capacitación de los trabajadores de la factoría y garantizar el seguimiento de esta, tema que se detallará a continuación.

Es de vital importancia para la UCI lograr la calidad de sus productos a través de los integrantes del equipo de desarrollo en los proyectos de SWE, es por eso que para tener un personal productivo es indispensable proporcionarles capacitación debido a la importancia que esta brinda al personal integrante del proyecto; si se imparte continuamente se generará en el estudiante (que es este caso sería un integrante del equipo de desarrollo) un interés hacia el conocimiento nuevo, además de su actualización, que serán los mejores aliados para el progreso de todo ser humano, porque a lo largo de la vida lo único que no pierde valor es el conocimiento acumulado, el cual no debe ser guardado ni privatizado sino puesto en práctica y socializado a las demás personas interesadas en el mismo, en pos de lograr producir software con mayor calidad que requieran de este conocimiento.

La capacitación dependerá en gran medida de las herramientas que se vayan a emplear en el desarrollo del proyecto, es decir, si se utiliza Software Propietario los integrantes del proyecto recibirán los cursos correspondientes a las herramientas para este tipo de software y si por el contrario se decide utilizar Software Libre la capacitación se orientará en este sentido.

Guerra y González en su Trabajo de Diploma titulado “Estrategia de Transferencia del Modelo de Factoría de Software aplicando inteligencia para Contenidos Educativos” en la entidad Centro de Inteligencia plantean que para capacitar al personal se deben realizar talleres sobre el Modelo de Factoría que se va a transferir, cursos sobre los lenguajes HTML, CSS, JavaScript y opcionalmente XML, breves cursos de Proceso de Software en Equipo (TSP) así como talleres sobre el trabajo con los instrumentos del PSP.

Después de tratado y estudiado a profundidad el tema de la capacitación en correspondencia con su gran ayuda para el proceso de producción que anteriormente se expuso y teniendo en cuenta la política seguida en la UCI, en la que todos los estudiantes deben estar vinculados a los proyectos productivos, se decide realizar algunos cambios.

Para el caso que se utilicen herramientas propietarias para el proceso de desarrollo la capacitación quedará estructurada de la siguiente forma:

- Incorporar los cursos de ActionScript y PHP a los cursos que tratan la temática de los lenguajes a utilizar antes mencionados. (Ver Anexo 8 y 9).

- Adicionar cursos que traten sobre las siguientes herramientas:
 - Dreamweaver / IDE. (Ver Anexo 10).
 - Macromedia Flash 8. (Ver Anexo 11).
 - CorelDRAW Graphics Suite. (Ver Anexo 12).
 - Macromedia Fireworks 8. (Ver Anexo 13).

Por su parte si se decide orientar el desarrollo del proyecto con herramientas de tipo libre la estructura de la capacitación será de la siguiente forma:

- Adicionar los cursos sobre los lenguajes ActionScript y PHP. (Ver Anexo 8 y 9).
- Incorporar el curso sobre la herramienta libre GIMP. (Ver Anexo 14).
- Realizar talleres sobre las herramientas Mtasc, SWFTools y SWFMill.

Teniendo en cuenta todo lo antecedido con respecto a la capacitación se llega a la conclusión que debe ser un proceso que se inicie con los estudiantes de nuevo ingreso en el proyecto, independientemente de que cuenten con experiencia previa o no. Es decir, los cursos se impartirán a todos los integrantes del proyecto para garantizar los conocimientos necesarios en los estudiantes y así el éxito del producto final.

2.4.5 Aplicación de la Fase de Aprendizaje al Proceso de Transferencia.

En esta fase se determina si se cumplieron los objetivos especificados y se plantean las recomendaciones para el siguiente plan de mejora.

Seguidamente se muestran los objetivos que se identificaron en la fase de establecimiento, los cuales fueron cumplidos al 100%.

- Mejorar el Proceso de Transferencia del Modelo de Factoría de Software.
- Mejorar la entidad Gestión de Proyecto.
- Mejorar la entidad Persona.
- Mejorar la entidad Bases Tecnológicas.
- Mejorar la entidad Centro de Inteligencia.
- Perfeccionar la Gestión de la Calidad.

- Reformar los Roles de los integrantes del proyecto.
- Adicionar nuevos Instrumentos del PSP a utilizar.
- Modernizar y adicionar nuevas herramientas.
- Reestructurar la capacitación del personal que integra el proyecto.

Debido a que esta fase es para aprender se recomienda efectuar una nueva evaluación para determinar las nuevas necesidades y fortalezas que servirán de entrada al nuevo plan de acción, delimitando nuevas áreas de mejora.

2.5 Conclusiones.

En este capítulo se proporciona una explicación detallada de cómo aplicar cada una de las fases del Modelo Ideal a la Estrategia de Transferencia del Modelo de Factoría de Software, donde se definen áreas de mejoras entre las cuales se encontraban: Gestión de Proyecto, el Personal, Bases Tecnológicas y Centro de Inteligencia, que fueron en las que se centró principalmente el proceso de mejora.

La propuesta de solución expuesta puede ser utilizada en cualquier proyecto de SWE en la UCI siempre y cuando en el mismo se implante la estrategia de transferencia, garantizando de esta forma obtener un proceso de desarrollo de software eficiente y con calidad una vez que sea aplicado el modelo de mejora a dicho proceso.

Capítulo 3: Validación de la Propuesta

3

3.1 Introducción.

En este capítulo se validará técnicamente la propuesta mediante el método de experto, que permite la evaluación cuantitativa de criterios previamente definidos, que consiente en realizar un estudio de expertos para determinar si se acepta o no la propuesta analizada.

3.2 Guía para validar la propuesta.

Para el desarrollo de la validación se realizaron un conjunto de pasos que se muestran a continuación:

1. Se elaboran los criterios de evaluación de acuerdo a las características de la propuesta y se organizan en cuatro grupos.

Grupo No. 1: Criterios de mérito científico.

1. Valor científico de la propuesta.
2. Calidad de la investigación.
3. Aporte científico.
4. Aptitud científica y eficacia de los investigadores.

Grupo No. 2: Criterios de implantación.

5. Necesidad de empleo de la propuesta.
6. Lograr la mejora del proceso de transferencia.
7. Aceptación de la propuesta por los ingenieros de software.

Grupo No.3: Criterios de flexibilidad.

8. Adaptabilidad en cualquier proyecto de SWE, independientemente del tipo de software que se utilice.
9. Capacidad del Modelo Ideal de lograr procesos con calidad.

10. Uso de los roles, instrumentos y herramientas necesarias para obtener un producto final eficiente.

Grupo No 4. Criterios de impacto.

11. Resultado en los proyectos que desarrollan SWE.

12. Organización en la mejora del proceso de transferencia en los proyectos de SWE.

13. Posible aplicación de la propuesta.

2. Se establece a cada grupo de criterios un peso relativo de acuerdo al porcentaje que representa cada grupo del total y los intereses a evaluar.

Grupo No. 1.....30

Grupo No. 2..... 25

Grupo No.3..... 20

Grupo No 4..... 25

3. Se seleccionan 7 expertos teniendo en cuenta su especialidad, grado científico y currículo.

4. Se les entrega a los expertos la propuesta para que estudien el tema a evaluar y dos modelos, uno para que valoren el peso relativo de cada criterio, teniendo en cuenta que la suma del valor dado por parte de los expertos a cada criterio de un grupo no exceda del peso relativo asignado a este y otro para realizar una evaluación cuantitativa de cada criterio con una escala de 1-5. (Ver Anexo 15 y 16).

5. Después de recibir los valores del peso relativo de cada criterio se construye la siguiente tabla.

Tabla 3 Resultado del trabajo de los expertos.

G	C/E	E1	E2	E3	E4	E5	E6	E7	Ep
30	C1	9	8	10	8	9	9	10	9
	C2	7	9	8	8	7	8	7	7.71428571
	C3	8	6	7	7	8	7	7	7.14285714

	C4	6	7	5	7	6	6	6	6.14285714
25	C5	8	7	8	9	7	9	8	8
	C6	10	8	9	8	10	7	9	8.71428571
	C7	7	9	8	8	8	9	8	8.14285714
20	C8	7	6	6	7	8	7	8	7
	C9	7	6	7	5	6	6	5	6
	C10	6	8	7	8	6	6	7	6.85714286
25	C11	8	8	9	7	9	7	10	8.28571429
	C12	10	9	8	8	10	9	7	8.71428571
	C13	7	8	8	9	6	9	7	7.71428571
T		100	99	100	99	100	99	99	53.5384615

6. Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2).

Para esto se sigue el procedimiento siguiente:

- Sea C el número de criterios que van a evaluarse y (E) el número de expertos que realizan la evaluación.
- Para cada criterio se determina (ΣE) que representa la sumatoria del peso dado por cada experto, (E_p) que representa la puntuación promedio de cada criterio y (ΔC) que representa la diferencia entre (ΣE) y ($M\Sigma E$).
- Se calcula el peso medio de cada criterio ($M\Sigma E$) y se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión. (Ver Tabla 4).
 - $S = \Sigma (\Sigma E - \Sigma \Sigma E / C)^2$

Tabla 4 Cálculo de la Dispersión (S) para hallar la concordancia entre los expertos.

C1	ΣE	$\Sigma E/C$	$\Sigma E - \Sigma \Sigma E/C$	$(\Sigma E - \Sigma \Sigma E/C)^2$
	63	4.846153	9.461545	89.5208338
C2	54	4.153846	0.461545	0.21302379
C3	50	3.846153	-3.538455	12.5206638
C4	43	3.307692	-10.538455	111.059034
C5	56	4.307692	2.461545	6.05920379
C6	61	4.692307	7.461545	55.6746538
C7	57	4.384615	3.461545	11.9822938
C8	49	3.76923	-4.538455	20.5975738
C9	42	3.230769	-11.538455	133.135944
C10	48	3.692307	-5.538455	30.6744838
C11	58	4.461538	4.461545	19.9053838
C12	61	4.692307	7.461545	55.6746538
C13	54	4.153846	0.461545	0.21302379
$\Sigma \Sigma E/C$	–	53.538455	–	–
$S = \Sigma (\Sigma E - \Sigma \Sigma E/C)^2$	–	–	–	547.230769

Conociendo la dispersión se calcula el coeficiente de concordancia de Kendall

$W = S / E^2 (C^3 - C) / 12$ y con los datos obtenidos de ese cálculo se prosigue a determinar el Chi cuadrado real $X^2_{real} = E (C-1) W$. (Ver Tabla 5).

Tabla 5 Cálculo de concordancia.

Expertos/Criterios	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇	∑E	Ep	ΔC	ΔC ²
C1	9	8	10	8	9	9	10	63	9	9	81
C2	7	9	8	8	7	8	7	54	7.71428571	0	0
C3	8	6	7	7	8	7	7	50	7.14285714	-4	16
C4	6	7	5	7	6	6	6	43	6.14285714	-11	122
C5	8	7	8	9	7	9	8	56	8	2	4
C6	10	8	9	8	10	7	9	61	8.71428571	7	49
C7	7	9	8	8	8	9	8	57	8.14285714	3	9
C8	7	6	6	7	8	7	8	49	7	-5	25
C9	7	6	7	5	6	6	5	42	6	-12	144
C10	6	8	7	8	6	6	7	48	6.85714286	-6	36
C11	8	8	9	7	9	7	10	58	8.28571429	4	16
C12	10	9	8	8	10	9	7	61	8.71428571	7	49
C12	7	8	8	9	6	9	7	54	7.71428571	0	0
C13	9	8	10	8	9	9	10	63	9	9	81
DC	100	99	100	99	100	99	99	696	108.428571	3	632
M ∑E	54										
W	0.06135										
X ²	5.1534										
X ² _(α, c-1)	26.2170										

El Chi cuadrado calculado se compara con el obtenido de las tablas estadísticas y de esta forma se obtiene la siguiente conclusión. Si se cumple la siguiente condición:

$$X_{\text{real}}^2 < X_{(\alpha, c-1)}^2$$

$$5.1534 < 26.2170$$

Se puede decir que existe concordancia en el trabajo de los expertos.

7. Posteriormente se identifica el peso relativo de cada criterio (P) y se calcula el Índice de Aceptación (IA) de la propuesta.

Para esto se sigue el procedimiento siguiente. (Ver Tabla 6).

- Conociendo el número de expertos que realizan la evaluación (E) y la sumatoria de las puntuaciones de cada criterio ΣC se puede calcular el peso de cada criterio (P).
- Conociendo el peso de cada criterio (P) y la calificación dada por los evaluadores (c) en una escala de 1-5 se puede calcular el valor de $P \times c$.
- Para recoger la calificación dada por los expertos a cada uno de los criterios se definió un modelo. (Ver Anexo 17)
- Con el valor anterior se calcula el Índice de Aceptación del proyecto (IA).
 - $IA = P \times C / 5$

Tabla 6 Calificación de cada criterio.

Criterios	Calificación (c)					P	P x c
	1	2	3	4	5		
C1					X	0.09	0.45
C2					X	0.077	0.385
C3					X	0.071	0.355
C4					X	0.061	0.305
C5				X		0.08	0.32
C6				X		0.087	0.348
C7					X	0.081	0.405

C8				X		0.07	0.28
C9					X	0.06	0.3
C10					X	0.068	0.34
C11					X	0.082	0.41
C12				X		0.087	0.348
C13					X	0.077	0.385
Total							4.631
IA	0.9262						

8. Se determina la probabilidad de éxito de la propuesta, para esto se ubica el Índice de Aceptación (IA) calculado anteriormente en rangos que ya están predefinidos, en dependencia de donde se ubique, será la probabilidad de éxito que tenga la propuesta.

El Índice de Aceptación calculado es 0.9262.

Rangos predefinidos de Índice de Aceptación.

IA > 0,7 Existe alta probabilidad de éxito.

0,7 > IA > 0,5 Existe probabilidad media de éxito.

0,5 > IA > 0,3 Probabilidad de éxito baja.

0,3 > IA Fracaso seguro.

Por lo que existe alta probabilidad de éxito.

3.3 Conclusiones.

En este capítulo se evaluó la propuesta de solución mediante el método multicriterio; el cual permitió analizar los criterios de cada uno de los expertos y determinar el índice de aceptación que tiene el presente trabajo, obteniéndose concordancia en el mismo y una alta probabilidad de éxito de ser aplicado el Modelo Ideal al Proceso de Transferencia del Modelo de Factoría de Software en los proyectos de SWE en la UCI.

Conclusiones

En el presente trabajo se efectuó un estudio de los principales modelos de mejora para los procesos de desarrollo de software actuales. Luego de este análisis se escoge el Modelo Ideal pues es el que más se adecúa a las necesidades del Proceso de Transferencia del Modelo de Factoría de Software que se pretendía mejorar, para una posterior aplicación de forma eficiente en los proyectos de Software Educativo en la UCI.

Siguiendo las pautas del Modelo Ideal se aplicaron cada una de sus fases al proceso de transferencia, centrándose en las áreas de mejora que presentaban mayor dificultad. Con este procedimiento para guiar la mejora de la estrategia de transferencia en los proyectos que desarrollan SWE, se logra dar cumplimiento al objetivo general de la investigación.

Finalmente se realizó la evaluación técnica de la propuesta mediante el método de experto, en el cual se obtuvo una alta probabilidad de éxito, de ser aplicado en proyectos productivos que desarrollen SWE, lo que implica desde el punto de vista teórico, el cumplimiento de la idea a defender planteada.

Recomendaciones

Con el objetivo de mejorar la propuesta planteada en esta investigación, se recomienda:

1. Realizar una prueba piloto en proyectos donde se aplique el procedimiento propuesto en la investigación para obtener resultados reales.
2. Redefinir el modelo propuesto para otra línea de producción.
3. La aplicación de modelos como este deben ser publicadas para incitar a otros productores a realizar el mismo procedimiento en sus producciones de software.
4. Efectuar otro ciclo del Modelo Ideal para identificar nuevas áreas de mejora que presenten dificultades.
5. Utilizar herramientas de monitoreo continuo que permitan evaluar los procesos.

Referencias Bibliográficas

A.CALVO-MANZANO, J.; G. C. AGUSTÍN, et al. A Software Process Improvement Solution for Small and Medium-Size Enterprises, 2006. [Disponible en:

<http://www.bax.com.br/teaching/courses/principiosSI/workshop-CMMinPMEsanais/Proceedings%20of%20the%20First%20International%20Research%20Workshop%20for%20Process%20Improvement%20in%20Small%20Settings-2005-p200-276.pdf>.

ACEVEDO, A., ZAMUDIO. Un Análisis de la Transferencia y apropiación del conocimiento en la investigación de Universidades Colombianas. Investigación y Desarrollo, 2005. Vol. 13: 128-157. Citado por GUERRA, Z. C. and L. H. GONZÁLEZ. Estrategia de Transferencia de un Modelo de Factoría de Software aplicando Inteligencia para contenidos Educativos., 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0722_07.pdf.

BADILLO, R. G. Un concepto epistemológico de modelo para la didáctica de las ciencias experimentales, 2004. [Disponible en

http://www.saum.uvigo.es/reec/volumenes/volumen3/Numero3/ART4_VOL3_N3.pdf.

B. McFeeley, IDEAL: A User's Guide for Software Process Improvement, tech. report CMU/SEI-96-HB-001, Software Eng. Inst., 1996. Citado por GARCÍA, J.; A. D. AMESCUA, et al. TOP 10 de factores que obstaculizan la mejora de los procesos de verificación y validación en organizaciones intensivas en software, 2006. [Disponible en: <http://www.ati.es/IMG/pdf/GarciaGuzmanNum2Vol2.pdf>.

CASAÑOLA, Y. T. Apuntes sobre Modelo Ideal, 2007. [Disponible en: <http://calidadsoft.prod.uci.cu/articulos/IDEAL%201.0.pdf>.

CATALDI, Z. Metodología de diseño, desarrollo y evaluación de Software Educativo 2000. [Disponible en: <http://www.fi.uba.ar/laboratorios/lsi/cataldi-tesisdemagistereninformatica.pdf>.

Cusumano, M A. (1989). Software Factory: A Historical Interpretation. IEEE Software, (Vol. 6, No. 2). pp. 23-30. 2006. Citado por Negrín, Y. M. (2007, 16/2/2008). "Modelo de Factoría de Software para la organización de Proyectos Productivos en el Instituto Politécnico "Abel Santamaría"." [Disponible en: http://bibliodoc.uci.cu/TD/TD_0247_07.pdf.

ESCORIAL, J. S. Calidad del Software, 2007. [Disponible en: <http://www.infor.uva.es/~jsalama1/calsoft.html>.

ESPAÑOLA, D. D. L. L. Diccionario de la lengua española. Madrid, 2005. p.

Española, D. d. l. l. (2005). Diccionario de la lengua española. Madrid. Citado por Guerra, Z. C. and L. H. González. (2007). "Estrategia de Transferencia de un Modelo de Factoría de Software aplicando Inteligencia para contenidos Educativos." [Disponible en: http://bibliodoc.uci.cu/TD/TD_0722_07.pdf

ESTRADA, A. F. El proceso de enseñanza aprendizaje y la vinculación universidad-empresa, una experiencia en una Universidad Cubana, 2002. [Disponible en: <http://www.somece.org.mx/memorias/2002/Grupo4/Febles.doc>.

GALLI, R. Introducción y ventajas del Software Libre, 2004. [Disponible en: <http://bulma.net/~gallir/BULMA/campos2004.pdf>.

GARCÍA, Z. N. and A. G. LANDEIRO. Entidad Proceso del Modelo de Factoría de Software aplicando Inteligencia, 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0716_07.pdf.

GONZÁLEZ, A. H. and Y. R. MARTÍNEZ. Desarrollo y Mantenimiento de una Distribución de Linux 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0129_06.pdf.

GREMBA, J. and T. MYERS. El modelo IDEAL (SM): Una guía práctica para la mejora, 1997. [Disponible en: <http://www.sei.cmu.edu/ideal/ideal.bridge.html>.

GUERRA, Z. C. and L. H. GONZÁLEZ. Estrategia de Transferencia de un Modelo de Factoría de Software aplicando Inteligencia para contenidos Educativos., 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0722_07.pdf.

HUMPHREY, W. S. Introducción al Proceso Software Personal, 2001. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg03058.pdf>.

HURTADO, J. A. Hacia una Línea de Procesos ágiles Agile SPsL, 2005. [Disponible en: <http://64.233.169.104/search?q=cache:l61ijCmUdWIJ:www.dcc.uchile.cl/~cecilia/papers/AgileSPsL.pdf+hacia+una+linea+de+procesos+agiles&hl=es&ct=clink&cd=1&gl=cu>.

JACOBSON, I.; G. BOOCH, et al. El Proceso Unificado de Desarrollo de Software, 2000. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

JOSÉ AUGUSTO FABRI, M. D. M. S., MARCELO SCHNECK DE PAULA, IVANIR COSTA, ELIANE D'IPPOLITO ElabTI: Um ambiente real e replicável de produção de software. Brasil

Escola Politecnica USP, 2004. p. Citado por GUERRA, Z. C. and L. H. GONZÁLEZ. Estrategia de Transferencia de un Modelo de Factoría de Software aplicando Inteligencia para contenidos Educativos., 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0722_07.pdf].

Li, C.; Li, H.; Li, M. A (2001). Software Factory Model Based on ISO 9000 e CMM for Chinese Small Organization. Second Asia-Pacific Conference on Quality Software (APAQS'01). Hong Kong. December. 2006. Citado por Negrín, Y. M. (2007, 16/2/2008). "Modelo de Factoría de Software para la organización de Proyectos Productivos en el Instituto Politécnico "Abel Santamaría"." [Disponible en: http://bibliodoc.uci.cu/TD/TD_0247_07.pdf].

MARQUÈS, P. El Software Educativo, 2005. [Disponible en: http://www.lmi.ub.es/te/any96/marques_software/].

MATURELL, A. S. Análisis de un IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar Software Educativo en formato multimedia., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

MÓNACO, S. D. and S. VITALI. El Modelo IDEAL para implementar CMMI 2005. [Disponible en: <http://rosario.sadio.org.ar/index.php?name=Downloads&req=getit&lid=19>].

MONTSERRAT, C. J.; G. H. W. GUADALUPE, et al. Software libre vs Software Propietario. Ventajas y desventajas, 2006. [Disponible en: <http://www.softwarelibre.cl/drupal//files/32693.pdf>].

NEGRÍN, Y. M. Modelo de Factoría de Software para la organización de Proyectos Productivos en el Instituto Politécnico "Abel Santamaría", 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0247_07.pdf].

PICHACO, A. M. Enginyeria del Software III L'Orientació a processos en el desenvolupament de software. Avaluació i millora d'aquests processos., 2007a. ---. Enginyeria del Software III Sessió 6. La millora dels processos de software a les petites i mitjanes empreses. El projecte QuaSAR, 2007b. [Disponible en: http://dmi.uib.es/~dmiamp/ESIII/0708_ESIII_SPI_Tema5.pdf].

PINO, F. J.; J. C. VIDAL, et al. Modelo para la Implementación de Mejora de Procesos en Pequeñas Organizaciones Software, 2007. [Disponible en: <http://www.sistedes.es/sistedes/pdf/2007/JISBD-07-pino-procesos.pdf>].

PYHÄJÄRVI, M. SPICE-International Standard for Software Process Assessment, 2004. [Disponible en: <http://www.cs.helsinki.fi/u/paakki/Pyhajarvi.pdf>].

QUINN, J. B. Definición de la Estrategia 2006. [Disponible en: http://www.persys.com.mx/secciones/estrategias_definicion.html].

RAMOS, R. R. Modelo de evaluación del proceso de desarrollo del Software Educativo., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

RINCÓN, C. and S. RODRÍGUEZ. Mejoramiento del Proceso de Planeación y Seguimiento de Proyectos de Desarrollo de Software en Pequeñas Empresas, 2006. Disponible en: http://qualdev.uniandes.edu.co/wikiProcess/doku.php?id=research:tesis_ser-rodr_c-rincon.

S.PRESSMAN, R. Ingeniería de Software. Un enfoque práctico, 2002. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg02689.pdf>].

SAGRARIO., A. Mejora de procesos: La evolución natural., 2005. [Disponible en: <http://www.als-es.com/recursos/articulos/mejora-procesos-cmmi.pdf>].

SANZ, L. F. and J. J. CUADRADO-GALLEGO. Revista Española de Innovación, Calidad e Ingeniería del Software, 2005. [1]. Disponible en: <http://www.ati.es/IMG/pdf/AntoniaMasNum2Vol1.pdf>.

Scott, L. et Al. Practical Software Process Improvement - The IMPACT. Project Proceedings of the 13th Australian Software Engineering Conference. IEEE 2001. Citado por Hurtado, J. A. (2005, 22/11/2007). "Hacia una Línea de Procesos Agiles Agile SPsL." from <http://64.233.169.104/search?q=cache:l61ijCmUdWIJ:www.dcc.uchile.cl/~cecilia/papers/AgileSPsL.pdf+hacia+una+linea+de+procesos+agiles&hl=es&ct=clnk&cd=1&gl=cu>.

TANTARA INC. Software process improvement & related standards/models. 2001. from http://www.tantara.ab.ca/a_stds.htm. Citado por Hurtado, J. A. (2005, 22/11/2007). "Hacia una Línea de Procesos ágiles Agile SPsL." from <http://64.233.169.104/search?q=cache:l61ijCmUdWIJ:www.dcc.uchile.cl/~cecilia/papers/AgileSPsL.pdf+hacia+una+linea+de+procesos+agiles&hl=es&ct=clnk&cd=1&gl=cu>.

WELCH, J. F. Definición de la Estrategia 2006. [Disponible en: http://www.persys.com.mx/secciones/estrategias_definicion.html].

Bibliografía

AGÜERO, D. N. Áreas del aseguramiento de la calidad., 2007. [Disponible en: <http://calidadsoft.prod.uci.cu/articulos/articulo-SQA.pdf>].

AGUIAR, H. P. and Y. B. TURRUELLES. Procedimiento para el control y aseguramiento de la calidad en la Gestión de Proyectos., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

ALÓN, L. D. and H. D. PÉREZ. Propuesta de Mecanismo de Gestión de Calidad Interna para el proyecto Registro y Notarías, 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

ANDRADE, R. A.; A. C. GÓMEZ, et al. DESDE ISO 9001 HACIA CMMI, PASOS PARA LA MEJORA DE LOS PROCESOS Y MÉTRICAS, 2007. [Disponible en: <http://www.aemes.org/rpm/descargar.php?volumen=4&numero=1&articulo=3>].

ARGUELLES, Y. C. Propuesta de aplicación de la norma ISO/IEC 15504 a los productos informáticos de la UCI, 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

BATISTA, M. L. R. Propuesta de procedimiento para el Aseguramiento de la Calidad del Software en los proyectos productivos de la facultad 7, 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

BAYOUMI, D.; K. RAJANIEMI, et al. Ideas brillantes, 2006. [Disponible en: [http://library.abb.com/GLOBAL/SCOT/scot271.nsf/VerityDisplay/ACB89993DA999825C1257199004F85E8/\\$File/54-57%202M639_SPA72dpi.pdf](http://library.abb.com/GLOBAL/SCOT/scot271.nsf/VerityDisplay/ACB89993DA999825C1257199004F85E8/$File/54-57%202M639_SPA72dpi.pdf)].

CASAÑOLA, Y. T. Evaluación teórica de la adopción del enfoque de Factorías de Software en la Universidad de las Ciencias Informáticas., 2006. [Disponible en: <http://www.intempres.pco.cu/Intempres2006/Intempres2006/Evaluacion%20de%20trabajos/Yaim%ED%20Trujillo%20Casa%F1ola%20P.pdf>].

CASAÑOLA, Y. T. Apuntes sobre Modelo Ideal, 2007. [Disponible en: <http://calidadsoft.prod.uci.cu/articulos/IDEAL%201.0.pdf>].

DUMENIGO, D. A. M. Análisis de herramientas libres para el desarrollo de productos educativos multimedia, 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

FERNÁNDEZ, D. R. Análisis y Gestión del desarrollo de software, 2007. [Disponible en: http://wainu.ii.uned.es:8081/WAINU/ing_informatica/cuarto/AGDS/apuntes/AGDS%20Tema%207.pdf].

FERNÁNDEZ, L. L. and O. D. IZQUIERDO. Procedimiento para la aplicación de un modelo de procesos que guíe la producción de software de la Facultad 3., 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0253_07.pdf].

KIMBROUGH, T. and L. LEVINE. The IDEAL sm Transition Frameworks-Speeding Managed Change, 1997. [Disponible en: <http://www.sei.cmu.edu/ideal/ideal-presentation.pdf>].

LASTRE, J. R. and Y. S. RODRÍGUEZ. Propuesta de Modelo de Calidad para Portales Web., 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0894_07.pdf].

LUZÚA, Y. B. Propuesta de un Proceso de Selección de Roles y Personal con sus Niveles de Competencia para Proyectos Multimedia., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

MARTÍNEZ, N. C. Estrategia de implantación del Modelo de Factoría de Software aplicando Inteligencia para el Polo Productivo Gestión de Proyecto., 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0243_07.pdf].

MATURELL, A. S. Análisis de un IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar Software Educativo en formato multimedia., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

MCFEELEY, B. IDEALSM: A User's Guide for Software Process Improvement, 1996. [Disponible en: <http://www.sei.cmu.edu/pub/documents/96.reports/pdf/hb001.96.pdf>].

MEDEROS, L. B. and A. L. KULIKOV. Definición de la entidad Persona para el Modelo de Factoría de Software aplicando inteligencia, 2007. [Disponible en: http://bibliodoc.uci.cu/TD/TD_0498_07.pdf].

MEJÍA, C. M. Bases para el Éxito en la Mejora de Procesos de SW, 2005. [Disponible en: <http://www.sg.com.mx/sg07/presentaciones/Mejora%20de%20procesos/SG07.P01.Bases%20para%20el%20exito%20en%20mejora%20de%20procesos.pdf>].

PÉREZ, V. M. Estudio sobre Herramientas de Diseño en Software Libre., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

- PINO, F. J.; F. GARCÍA, et al. Herramienta de Soporte a la Valoración Rápida de Procesos Software, 2007a. [Disponible en: <http://ieeexplore.ieee.org/iel5/9907/4378505/04378509.pdf?isnumber=4378505&prod=JNL&arnumber=4378509&arSt=218&ared=223&arAuthor=Pino%2C+Francisco+J.%3B+Garcia%2C+Felix%3B+Piattini%2C+Mario>].
- PORTILLO, N. P. El Mito sobre CMMI en Pequeñas Organizaciones, 2007. [Disponible en: <http://www.sg.com.mx/sg07/presentaciones/Mejora%20de%20procesos/SG07.P05.CMMI%20en%20pequeñas%20orgs.pdf>].
- QUIÑONES, E. Modelos de Calidad de Software y Software Libre, 2007. [Disponible en: http://www.eqsoft.net/presentas/modelos_de_calidad_y_software_libre.pdf].
- RIBEIRO, E.; E. SALIBA, et al. A Software Process Improvement Solution for Small and Medium-Size Enterprises (5.2), 2005. [Disponible en: http://www.bax.com.br/teaching/courses/principiosSI/seminario/Seminario_01_trab_Ultimo_V20_03.ppt].
- ROJAS, O. M. Propuesta para la Gestión de Proyectos durante el proceso de desarrollo de software en proyectos productivos de gestión en la UCI., 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].
- ROMERO, C. I. G. El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicana de Software, 2001. [Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/capitulo4.pdf].
- RÚA, E. H. C. Mejora de la calidad del software en entorno de microempresas de TI, 2007. [Disponible en: <http://is.ls.fi.upm.es/doctorado/Trabajos20062007/Caballero.pdf>].
- RUBIO, S. E. D. BOOTSTRAP. El Estándar Europeo para Evaluación y Mejoras de Procesos de Desarrollo de Software, 1997. [Disponible en: <http://www.certum.com/Publicaciones/Bootstrap.pdf>].
- RUIZ, J. H. Propuesta de documentación para los Procesos de Calidad del Software, 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].
- SAMÓN, R. P. Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas (UCI). 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

VALERIS, I. H. Propuesta de Control para la Gestión de la Calidad del Software en el Proceso Productivo UCI, 2007. [Disponible en: <http://biblioteca.uci.cu/sbd/biuci/index.html>].

VELTHUIS, M. P. COMPETISOFT: Mejora de procesos para Pymes, 2006. [Disponible en: <http://www.calidaddelsoftware.com/documentos/III%20Semana%20CMMI/Ponencias%20III%20Semana%20CMMI/UCLM%20-%20III%20SemanaCMMIFINAL.pdf>].

ANEXOS

Anexo 1: Ejemplo de la portada del Cuaderno del Ingeniero.

Cuaderno número: <u> 1 </u>	
 Cuaderno de ingeniería Nombre de la Empresa o Universidad 	
Nombre del Ingeniero	<u> Jane Doe </u>
Teléfono/correo electrónico	<u> jd@dbxyz.edu </u>
Fecha de apertura	<u> 9/9/96 </u>
Fecha de cierre	<u> </u>

Anexo 4: Cuaderno de Registro de Defectos.

Estudiante _____					Fecha _____	
Profesor _____					Programa # _____	
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						
Fecha	Número	Tipo	Introducido	Eliminado	Tiempo de corrección	Defecto corregido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Descripción: _____						

Anexo 6: Lista de compromisos.

Nombre _____ Fecha _____

Profesor _____ Clase _____

Fecha comprometida	Compromiso	¿Con quién?	Horas	Fecha de	Consigo
Semanal					
Otros					

Anexo 7: El Plan del Proyecto.

Estudiante _____				Fecha _____
Programa _____				Programa # _____
Profesor _____				Lenguaje _____

Resumen	Plan	Real	Hasta la Fecha		
Minutos/LOC	_____	_____	_____		
LOC/Hora	_____	_____	_____		
Defectos/KLOC	_____	_____	_____		
Rendimiento	_____	_____	_____		
V/F	_____	_____	_____		
Tamaño Programa (LOC):					
Total Nuevo & Cambiado	_____				
Tamaño Máximo	_____				
Tamaño Mínimo	_____				
Tiempo por Fase (min.)	Plan	Real	Hasta la Fecha	% Hasta la Fecha	
<i>Planificación</i>	_____	_____	_____	_____	
<i>Diseño</i>	_____	_____	_____	_____	
<i>Codificación</i>	_____	_____	_____	_____	
<i>Revisión del código</i>	_____	_____	_____	_____	
<i>Compilación</i>	_____	_____	_____	_____	
<i>Pruebas</i>	_____	_____	_____	_____	
<i>Postmorten</i>	_____	_____	_____	_____	
Total	_____	_____	_____	_____	
Tiempo Máximo	_____				
Tiempo Mínimo	_____				
Defectos introducidos	Plan	Real	Hasta la Fecha	% Hasta la Fecha	Def./Hora
<i>Planificación</i>	_____	_____	_____	_____	_____
<i>Diseño</i>	_____	_____	_____	_____	_____
<i>Codificación</i>	_____	_____	_____	_____	_____
<i>Revisión del código</i>	_____	_____	_____	_____	_____
<i>Compilación</i>	_____	_____	_____	_____	_____
<i>Pruebas</i>	_____	_____	_____	_____	_____
Total	_____	_____	_____	_____	_____
Defectos eliminados	Plan	Actual	Hasta la Fecha	% Hasta la Fecha	Def./Hora
<i>Planificación</i>	_____	_____	_____	_____	_____
<i>Diseño</i>	_____	_____	_____	_____	_____
<i>Codificación</i>	_____	_____	_____	_____	_____
<i>Revisión del código</i>	_____	_____	_____	_____	_____
<i>Compilación</i>	_____	_____	_____	_____	_____
<i>Pruebas</i>	_____	_____	_____	_____	_____
Total	_____	_____	_____	_____	_____

Anexo 8: Estructura del curso de ActionScript.

- Introducción al lenguaje ActionScript.
- Controlar objetos visuales con ActionScript.
 - Variable.
 - Nombres de propiedades e instancia.
 - Texto.
 - Trabaja con números y cadenas.

- Utilizar y escribir funciones.
 - Cargar contenido externo.
 - Creación de instancias desde la biblioteca.
 - Función para crear MovieClip.
- Comprensión y gestión de eventos.
 - Evento.
 - Controlar una línea de tiempo de MovieClip.
- Gestionar color, sonido y datos con clases.
 - Variables.
 - Crear matrices.
 - Trabajo con matriz.
 - Creación de un objeto genérico.
 - Transformación de objetos visuales mediante código.
 - Reproducción de sonido.
- Sentencias.
 - Sentencia de bucle.
 - Sentencia condicional.
- Animación con ActionScript.

Anexo 9: Estructura del curso de PHP.

- Introducción al PHP.
- Comentarios.
- Variables.
- Constantes.
- Operadores.

- Estructuras de control.
 - Condicional if.
 - Bucle while.
 - Bucle do while.
 - Bucle for.
 - El ciclo switch.
- Funciones.
- PHP orientado a objetos.
- Imagen, descripción y ejemplos.
- Creación de una imagen.
 - Uso del color.
 - Dibujo de líneas.
 - Dibujo de arcos y elipses.
 - Dibujo de rectángulos.
- Procesamiento de formularios.
- Cookies en PHP.
- Sesiones PHP.
- Funciones de FTP.

Anexo 10: Estructura del curso de Dreamweaver / IDE.

- Introducción a Dreamweaver.
 - Descripción de los principales elementos del entorno de trabajo de Dreamweaver (paneles y herramientas).
 - Utilización del sistema de ayuda.
- Crear sitios web.
 - Pasos a seguir para crear y configurar un sitio web.

- Diseño de páginas web.
 - Crear y configurar nuevas páginas web.
 - Trabajo con los colores.
 - Trabajo con las ventanas.
 - Utilización de reglas, guías y cuadrícula como ayuda para la organización de los objetos.
- Introducir texto.
 - Estudio del formato del texto (fuente, tamaño, color, alineación, sangría), los estilos CSS.
 - Creación listas (numeradas, no numeradas o de definición).
- Trabajo con vínculos.
 - Insertar hipervínculos en las páginas web.
 - Administración y modificación de hipervínculos en Dreamweaver.
- Insertar imágenes.
 - La creación de imágenes de sustitución, mapas de imagen y barras de navegación.
- Capas y tablas.
 - La utilización de capas, tablas y la utilización del modo de diseño, asignación del formato y incluir el contenido.
- División en marcos
 - Se explica el modo de trabajar con los marcos, así como la asignación de contenido y la creación de hipervínculos que trabajen con los marcos.
 - Utilización de plantillas.
- Formularios.
 - Creación y configuración de los formularios.
- Elementos multimedia.
 - Inserción de animaciones de Flash, vídeos de Flash, botones y textos Flash.
 - Trabajo conjunto entre Dreamweaver y Fireworks, como la creación de un álbum de fotos web o la inserción de HTML procedente de Fireworks en una página web diseñada con Dreamweaver.
- Publicar el sitio web.

- Explica los pasos a seguir para preparar el sitio web Dreamweaver y finalmente publicarlo en el servidor web.

Anexo 11: Estructura del curso de Macromedia Flash 8.

- Introducción a Macromedia Flash 8.
 - Novedades de la versión.
 - Sistemas de ayuda y soporte al aprendizaje.
- Herramientas y Propiedades.
 - La Barra de Herramientas.
 - El Inspector de Propiedades.
 - Herramientas de trazado.
 - Herramientas de relleno.
 - Herramienta de borrado.
- Herramientas y Paneles de Modificación.
 - Herramienta Selección y Subselección.
 - Intersección de trazados y rellenos.
 - La opción agrupar, separar y organizar.
 - Trazar una imagen de mapa de bits.
 - Transformaciones básicas.
 - Pivote.
 - Modificación de forma.
 - El panel alinear.
- Las Capas y sus Propiedades
 - Trabajo con capas.
 - Carpetas.

- Capas Guía.
- Máscaras.
- Trabajo con Texto.
 - Creación de bloques de texto.
 - El Inspector de Propiedades.
 - Propiedades básicas y avanzadas.
 - La opción Separar.
- Símbolos e Instancias
 - Tipos de símbolos.
 - Anidar símbolos.
 - El panel Biblioteca.
 - Efectos de instancia.
 - Abrir biblioteca externa.
- Animación en Flash.
 - La Línea de Tiempo.
 - Edición de fotogramas.
 - Animación fotograma a fotograma.
 - Interpolaciones.
 - Guía de movimiento.
 - Interpolación con máscaras.
 - Interpolación de efectos de instancia.
 - Combinación de animaciones.
 - Trabajo con escenas.
- Multimedia.

- Sonido en Flash.
- Propiedades del sonido.
- Video en Flash.
- Componentes para el control de video.
- Publicación.
 - Publicar una película Flash.
 - Formatos y opciones de exportación.

Anexo 12: Estructura del curso de CorelDRAW Graphics Suite.

- Introducción, trabajo con archivos.
- Conceptos generales y aplicaciones del Corel Draw Graphics Suite.
- Iniciar Corel DRAW y entorno de trabajo.
- La caja de herramientas y paneles internos.
- Ayuda en Corel DRAW y salir de Corel DRAW.
- Abrir dibujos e importar dibujos.
- Crear nuevos dibujos, guardar dibujos y exportar dibujos.
- Configuración de los dibujos, objetos y propiedades.
- Preparar página, uso de herramientas y calidad de visualización.
- Deshacer errores, dibujo de líneas, curvas y mano alzada.
- Unión automática, polígono y espiral.
- Opciones de relleno, texturas y asignación de contornos.
- Seleccionar objetos simples y mover a una posición.
- Duplicar, clonar y otras transformaciones.
- Formar y organizar objetos.
- Introducción y formatos de texto.

- Edición de nodos y conversión en curvas.
- Rectángulos redondeados, sectores y arcos.
- Modificación de líneas y curvas.
- Escalar, rotar nodos y cuchillo y borrador.
- Reordenar, agrupar y desagrupar objetos.
- Combinar, descombinar, alinear y distribuir objetos.
- Interceptar, recortar, soldar y trabajo con capas.
- Tipos, conversión y edición de textos.
- Más formatos de texto.
- Efectos especiales, base de datos gráfica, documentos para Internet e impresión.
- Aplicar efectos especiales.
- Datos asociados al objeto, trabajo con bitmaps.
- Utilidades gráficas del Corel Draw Graphics Suite.
- Introducción a:
 - Corel R.A.V.E.
 - Corel PHOTO-PAINT.

Anexo 13: Estructura del curso de Macromedia Fireworks 8.

- Introducción a Fireworks.
 - Se presenta el entorno de trabajo de la aplicación.
 - Trabajo general con sus numerosos paneles.
 - Utilización del sistema de ayuda del programa.
- Trabajo con documentos.
 - Creación y configuración de los documentos.
 - Modificar un documento existente y cómo recortar y rotar el lienzo.

- Dibujo de objetos.
 - Imágenes de mapa de bits y las imágenes vectoriales.
 - Herramientas de dibujo.
- Edición de objetos.
 - Técnicas de edición de objetos: cómo seleccionarlos, remodelar el trazado y aplicar transformaciones.
- Trabajo con texto.
 - Edición del texto, revisión ortográfica, aplicación de efectos y transformación del texto.
- Grupos, capas y símbolos.
 - Símbolo e instancia.
 - La distribución de los objetos en capas y la forma de trabajar con estos elementos.
- Mapas de bits.
 - Cómo importar, las distintas formas de seleccionar áreas de estas imágenes y la edición de los píxeles de las mismas.
 - Mejora o modificación de una imagen de mapa de bits, ajuste de luces y sombras, corrección de color, aplicación de filtros y otras herramientas de retoque.
- Mezclas, máscaras y filtros automáticos.
 - Utilización de los modos de mezcla para variar la transparencia o variación de colores entre dos objetos superpuestos.
 - Las máscaras de vectores y de mapa de bits y la aplicación de filtros automáticos a los objetos.
- Zonas interactivas y divisiones.
 - Asignación de vínculos a elementos del documento, así como la manera de dividirlo en distintas partes, para optimizar y exportar cada área por separado.
- Botones, rollovers y menús emergentes.
 - Creación de elementos dinámicos (como botones de acción, rollovers o menús emergentes).
- Creación de animaciones.
 - Técnica para crear animación, viendo las distintas posibilidades que brinda el programa.
- Automatización y HTML.

- Herramientas que facilitan el trabajo con la aplicación, como la búsqueda y reemplazo, los archivos de comandos y el procesamiento por lotes.
- Configuración del HTML generado por este programa.

Anexo 14: Estructura del curso de Gimp.

- Herramienta de selección lazo.
- Herramienta medida.
- Herramienta de selección varita mágica.
- Herramienta de selección mediante color.
- Herramienta blanquear o ennegrecer.
- Herramienta goma.
- Herramienta de mover.
- Herramienta de selección elíptica.
- Herramienta relleno.
- Herramienta de selección tijeras inteligentes.
 - Máscaras rápidas.
 - El cuadro de diálogos de paleta de colores.
 - El color.
 - Combinar capas.
- Herramienta gradiente.
 - Modo clarear.
 - Trazar.
 - Transparencia de capas.
 - Utilizando los filtros.
 - Máscara de capa.
- Herramienta de texto.
 - Modos de mezclas.
- Herramienta lupa.
- Herramienta pincel.
 - Degradados (gradiente).
 - Utilizar la lupa.

- Máscaras de capa y de canal con el gradiente.
- Herramienta recorte.
 - Sombrear textos.
 - Efectos de filtros sobre textos.
 - Efectos de luz.
- Herramienta transformación.
 - Barra de menú.
- Herramienta de selección rectangular.
- Herramienta simétrica.
- Herramienta lápiz.
 - Girar y arquear textos.
- Herramienta tiznar.
 - Combinar canales.
 - Máscaras de canal y la herramienta umbral.

Anexo 15: Modelo para la recogida de información referente al peso de los criterios.

Guía para informar el peso de los criterios.

Fecha de recepción...00/00/00.....

Fecha de entrega....00/00/00.....

Nombre y Apellidos del evaluador.....

Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:

Grupo No. 1.....30

Grupo No. 2..... 25

Grupo No.3..... 20

Grupo No 4..... 25

Para que el peso total asignado sea 100.

Grupo No 1: Criterios de mérito científico.

1. Valor científico de la propuesta.

Peso.....

2. Calidad de la investigación.

Peso.....

3. Aporte científico.

Peso.....

4. Aptitud científica y eficacia de los investigadores.

Peso.....

Grupo No 2: Criterios de implantación.

5. Necesidad de empleo de la propuesta.

Peso.....

6. Lograr la mejora del proceso de transferencia.

Peso.....

7. Aceptación de la propuesta por los ingenieros de software.

Peso.....

Grupo No 3: Criterios de flexibilidad.

8. Adaptabilidad a proyectos productivos de SWE independientemente del tipo de SWE que utilicen.

Peso.....

9. Capacidad del Modelo Ideal de lograr procesos con calidad.

Peso.....

10. Uso de los roles, instrumentos y herramientas necesarias para obtener un producto final eficiente.

Peso.....

Grupo No 4. Criterios de impacto.

11. Resultado en los proyectos que desarrollan SWE.

Peso.....

12. Organización en la mejora del proceso de transferencia en los proyectos de SWE.

Peso.....

13. Posible aplicación de la propuesta.

Peso.....

Anexo 16: Modelo para la recogida de información referente a la calificación de los criterios.

Fecha de recepción...00/00/00.....

Fecha de entrega.... 00/00/00.....

Nombre y Apellidos del Evaluador

Criterios de medida que se evalúan en una escala de 1 - 5

Grupo No 1: Criterios de mérito científico.

1. Valor científico de la propuesta.

Peso.....

2. Calidad de la investigación.

Peso.....

3. Aporte científico.

Peso.....

4. Aptitud científica y eficacia de los investigadores.

Peso.....

Grupo No 2: Criterios de implantación.

5. Necesidad de empleo de la propuesta.

Peso.....

6. Lograr la mejora del proceso de transferencia.

Peso.....

7. Aceptación de la propuesta por los ingenieros de software.

Peso.....

Grupo No 3: Criterios de flexibilidad.

8. Adaptabilidad a proyectos productivos de SWE independientemente del tipo de SWE que utilicen.

Peso.....

9. Capacidad del Modelo Ideal de lograr procesos con calidad.

Peso.....

10. Uso de los roles, instrumentos y herramientas necesarias para obtener un producto final eficiente.

Peso.....

Grupo No 4. Criterios de impacto.

11. Resultado en los proyectos que desarrollan SWE.

Peso.....

12. Organización en la mejora del proceso de transferencia en los proyectos de SWE.

Peso.....

13. Posible aplicación de la propuesta.

Peso.....

Categoría final del proyecto

___ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

___ Bueno: Novedad científica, resultados destacados.

___ Aceptable: Suficientemente bueno con reservas.

___ Cuestionable: No tiene relevancia científica y los resultados son malos.

___ Malo: No aplicable.

Valoración final.

Sugerencias del experto para mejorar la calidad del proyecto.

Elementos críticos que deben mejorarse.

GLOSARIO

ActionScript: es un lenguaje de programación orientado a objetos, utilizado en especial en aplicaciones Web animadas realizadas en el entorno Macromedia Flash, la tecnología de Macromedia para añadir dinamismo al panorama Web.

Calidad: Conjunto de propiedades inherentes a una cosa que permiten apreciarla como igual, mejor o peor que las restantes de su especie.

Capacitación: Es toda la acción organizada y evaluable que se desarrolla en un proyecto para modificar, mejorar y ampliar los conocimientos, habilidades y actitudes del personal produciendo un cambio positivo en el desempeño de sus tareas. El objeto es perfeccionar al trabajador en su puesto de trabajo.

Cliente: Son las personas para las cuales se elabora un producto determinado.

Estrategia: Conjunto de acciones que se llevan a cabo para lograr un determinado fin.

Factoría: Cualquier tipo de fábrica o industria, es decir, a cualquier tipo de instalación en la cual se produce la transformación de materias primas o productos semiterminados en otros productos.

Factoría de software: Una Factoría de Software es el sentido de producir con rapidez y calidad a través de procesos conocidos, repetibles y mejorables continuamente. Incorpora técnicas, metodologías y herramientas en el desarrollo del software, que mantiene una mejora continua de procesos y trae como resultados la industrialización en la producción de software.

Fase: Período de tiempo entre dos hitos principales de un proceso de desarrollo.

Gestión de proyectos: Es la aplicación de varios conocimientos, habilidades, herramientas y técnicas para planificar y satisfacer las actividades de un proyecto.

Herramientas: Software que se utiliza para automatizar las actividades definidas en el proceso.

Modelo: Arquetipo digno de ser imitado que se toma como pauta a seguir.

Modelo de Factoría de software: Es una forma de representar el enfoque de Factoría de Software.

Proceso: Es un conjunto de actividades o eventos que se realizan o suceden con un determinado fin.

Proceso de software: Es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto para transformar los requisitos de usuario en un producto.

Producto: Resultado concreto, observable y medible que surge como consecuencia del proceso, proyecto o experiencia desarrollada.

Proyecto: Es el conjunto de operaciones limitadas en el tiempo, de las cuales resulta un producto final.

Repositorio: Servidor o dispositivo donde encontramos almacenados los programas que incluye una distribución.

Rol: Es el papel que desempeña una persona o grupo de ellas en cualquier actividad. Una persona puede tener varios roles, así como un rol puede ser desempeñado por varias personas.

Software Educativo: Programas informáticos para facilitar los procesos de enseñanza y de aprendizaje.

SWF: Es un formato de archivo de gráficos vectoriales creado por el programa Flash de Adobe (antes Macromedia).

Transferencia tecnológica: Consiste en transferir el conocimiento adquirido para construir determinado producto, mediante mecanismos que establecen una estrategia de transferencia de conocimientos.

Usuario: Es la persona que utiliza o trabaja con algún objeto o que proviene de algún servicio público o privado, empresarial o profesional.

Siglas

BSD: Berkeley Software Distribution.

CMM: Capability Maturity Model.

CMMI: Capability Maturity Model Integration.

CNTI: Centro Nacional de Tecnologías de la Información.

GIMP: GNU Image Manipulation Program.

GNU: Es un acrónimo recursivo para “GNU No es Unix”.

GPL: General Public License.

IEC: International Electrotechnical Commission.

MSG: Grupo de Gestión de Manejo.

MTASC: Motion Twin Action Script Compiler.

ISO: International Organization for Standardization.

PSP: Personal Software Process o Proceso de Software Personal.

SEI: Software Engineering Institute.

SEPG: Grupo de Proceso de Ingeniería de Software.

SPI: Software Process Improvement.

SPICE: Software Process Improvement and Capability Determination.

SQA: Aseguramiento de la Calidad del Software.

SWE: Software Educativo.

TI: Tecnologías de la Información.

TIC: Tecnologías de la Información y las Comunicaciones.

TWG: Grupo de Trabajo Técnico.

UCI: Universidad de Ciencias Informáticas.

UML: Lenguaje Unificado de Modelado