

**Universidad de las Ciencias Informáticas
Facultad 8**



***“Estrategia de Transferencia del Modelo de
Factoría de Software aplicando Inteligencia para
Contenidos Educativos”***

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

Autores: Lidisvey Herrero González

Zilenys Corrales Guerra

Tutor: Ing. Yaimí Trujillo Casañola

Ciudad de La Habana, Cuba

Junio, 2007

“Año 49 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 _____.

Lidisvey Herrero González
Autor

Zilenys Corrales Guerra
Autor

Ing. Yaimí Trujillo Casoñola
Tutor

DATOS DE CONTACTO

Nombre y Apellidos: Ing. Yaimí Trujillo Casañola

Fecha de nacimiento: 20 de Octubre de 1981

Institución: Universidad de las Ciencias Informáticas

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 81102008414

Correo: yaimi@uci.cu

Situación laboral: Profesora Instructora, Departamento de la Especialidad

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

Ingeniera Informática del Instituto Superior Politécnico José Antonio Echeverría desde Julio del 2004. Al graduarse pasa a ser profesora de la Universidad de las Ciencias Informáticas, en la Disciplina de Ingeniería y Gestión de Software. Obtiene la categoría de Instructora en octubre del 2005 y actualmente está en el proceso de categorización optando por la categoría de Asistente. Ha impartido asignaturas como Introducción a la Programación, Programación I, Práctica Profesional, Interpretación de UML, Introducción al Enfoque de Factoría de Software, Ingeniería de Software I y II entre otras. Ha desarrollado proyectos de investigación, tiene 6 artículos publicados en memorias de eventos científicos y Sitios Web. Ha ejercido como tutora de varias tesis de grado, así como ponente y miembro de tribunales. Se encuentra cursando la maestría de Gestión de Proyectos Informáticos. Ha participado como ponente en eventos científicos nacionales e internacionales de la rama. Ha trabajado en proyectos productivos como Programadora, Analista, Diseñadora y Líder de proyecto obteniendo software utilizados en empresas del país y productos para la exportación.

“La inteligencia consiste no sólo en poseer el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica...”

Aristóteles

AGRADECIMIENTOS

A la universidad de excelencia, la UCI, por permitirnos ser parte de su brillante tropa futuro.

A nuestros padres por su apoyo, amor, confianza y dedicación.

A nuestros profesores por enseñarnos el camino correcto y forjarnos como excelentes profesionales.

A nuestra tutora por su ayudarnos y guiarnos en toda la investigación.

A Alcides, nuestro decano, por brindarnos sus conocimientos y permitirnos esclarecer nuestras ideas.

A Grethel, Mayuli, Yade, Alejandro y Gilbert, por ser nuestros mejores amigos. Compartimos juntos este largo camino y les debemos mucho de lo que hoy somos.

A todas las personas que nos han brindado su apoyo en la realización de esta investigación.

A mi hermanito Dany, por darme tu alegría, cariño y ayudarme a ser mejor.

A Maykel, por darme todo tu amor y hacerme parte de tu vida.

A Nani, por ser más que mi compañera de tesis, mi amiga y hermana.

A mis compañeros de estudio, por ofrecerme lindos recuerdos de mi carrera estudiantil.

A mi familia, y todos aquellos que confiaron en mi, y que de una forma u otra me ayudaron a realizar este sueño.

A mi prima Liniet, Israel, Katherine, Mayelín, Erasmo y mis tíos Nani, Esther, Gonzalo, Airán y Yipsy por su preocupación y cariño.

A Pedrito y sus padres por su ayuda en todo momento.

A Zile, por soportarme en todos estos años en los buenos y malos tiempos de la carrera.

A mis compañeros de grupo, especialmente las muchachitas por haber compartido momentos bonitos.

Zilenys

Lidisvey

DEDICATORIA

A mis padres, Danilo y Vivian, por confiar en mí, por todo su amor, por guiarme y apoyarme en los buenos y malos momentos de mi vida. Porque son mi razón de ser y les debo todo lo que soy.

Zilenys

A mis padres, Humberto y Mariela, por su amor incondicional y la confianza depositada en mí para superar todos los obstáculos.

A mis abuelitos, Remberto y Fela, por su educación y desvelo por que alcance todos mis sueños.

Lidisvey

RESUMEN

Desde hace varias décadas la industria cubana del software ha evolucionado de forma significativa, convirtiéndose en una fuente de ingresos para el país. La Universidad de las Ciencias Informáticas (UCI) promueve el desarrollo de la informática tanto en el ámbito nacional como internacional, mediante proyectos productivos que vincula estudiantes y profesores a la producción de software. El proyecto productivo CNTI – Contenidos Educativos tiene la misión de producir una gran cantidad de contenidos educativos para la empresa Sistemas Informáticos de Software SIS – Copextel y Centro Nacional de Tecnologías de la Información de Venezuela (CNTI). El objetivo de este trabajo es diseñar una estrategia de transferencia de un modelo de Factoría de Software que elimine las deficiencias existentes en los proyectos de este tipo. Para cumplir con el objetivo propuesto se utilizó la metodología de la investigación científica. El resultado más relevante del trabajo se refleja en la organización de los integrantes del proyecto. La estrategia definida sirvió para eliminar insuficiencias en este sentido aunque no se probó completamente.

PALABRAS CLAVE

Factoría de Software, Modelo de Factoría de Software, Estrategia de transferencia

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción	4
1.2 Factoría de Software	4
1.2.1 Definiciones de Factoría de Software.....	4
1.2.2 Aspectos importantes sobre las Factorías de Software.....	6
1.3 Modelos de Factoría de Software	9
1.3.1 Modelo basado en la norma ISO 9001 y CMM.....	9
1.3.2 Modelo Eureka.....	11
1.3.3 Modelo Clasificadorio.....	12
1.3.4 Modelo propuesto por Basili.....	14
1.3.5 Modelo Replicable.....	15
1.3.6 Modelo aplicando Inteligencia.....	16
1.3.7 Análisis comparativo de los modelos.....	17
1.4 Experiencia Internacional	18
1.5 Estrategia	20
1.5.1 Definiciones de Estrategia.....	20
1.5.2 Enfoques de las estrategias.....	21
1.5.3 Niveles de Estrategia.....	22
1.6 Transferencia tecnológica y del conocimiento	23
1.6.1 Transferencia de un Modelo de Factoría de Software.....	24
1.6.2 Estrategias de transferencia.....	25
1.7 Metodologías de desarrollo de Software	26
1.8 Estándares de Calidad	27
1.8.1 Capability Maturity Model Integration (CMMI).....	27
1.8.2 International Standard Organization (ISO).....	28
1.8.3 PSP (Personal Software Process) y TSP (Team Software Process).....	29
1.9 Tecnología multimedia	30
1.10 Tecnologías y plataformas de desarrollo actuales	31
1.10.1 Tecnologías para sitios Web.....	31
1.10.2 Plataformas de desarrollo Web.....	33
1.10.3 Selección de las herramientas a utilizar.....	35
1.11 Conclusiones	35
CAPÍTULO 2: ESTRATEGIA DE LA TRANSFERENCIA	37
2.1 Introducción	37
2.2 Proceso de desarrollo de software educativo	37

2.3 Métodos, procedimientos y técnicas utilizados	39
2.4 Estrategia de transferencia del modelo	41
2.4.1 Gestión de Proyecto	44
2.4.1.1 Procesos de gestión	44
2.4.1.2 Gestión del alcance	45
2.4.1.3 Gestión del tiempo	47
2.4.1.4 Gestión del costo	47
2.4.1.5 Gestión de la calidad	48
2.4.1.6 Gestión de recursos	49
2.4.1.7 Gestión de riesgos	50
2.4.2 Proceso de desarrollo	52
2.4.2.1 Elaboración	52
2.4.2.2 Prueba	54
2.4.2.3 Entrega	55
2.4.3 Centro de Inteligencia	55
2.4.3.1 Capacitación	57
2.4.4 Repositorio	58
2.4.5 Personas	58
2.4.5.1 Roles de la factoría	58
2.4.5.2 Instrumentos PSP para utilizar	61
2.4.6 Bases Tecnológicas	62
2.4.6.1 Tecnologías y herramientas	62
2.4.6.2 Mecanismos	63
2.5 Implantación del proceso de transferencia	63
2.6 Conclusiones	66
CAPÍTULO 3: RESULTADOS	67
3.1 Introducción	67
3.2 Resultados de las técnicas utilizadas	67
3.4 Observaciones sobre la prueba de la estrategia	68
3.4.1 Gestión de Proyecto	68
3.4.1.1 Gestión del alcance	68
3.4.1.2 Gestión del tiempo	68
3.4.1.3 Gestión del costo	69
3.4.1.4 Gestión de la calidad	69
3.4.1.5 Gestión de los recursos	69
3.4.1.6 Gestión de riesgos	69
3.4.2 Proceso de desarrollo	70
3.4.3 Centro de Inteligencia	71
3.4.4 Repositorio	71

3.4.5 Personas	71
3.4.6 Bases tecnológicas	72
3.5 Reporte de los resultados	72
3.6 Análisis de los riesgos de la transferencia	73
3.7 Conclusiones	74
CONCLUSIONES	75
RECOMENDACIONES	76
REFERENCIAS BIBLIOGRÁFICAS	77
BIBLIOGRAFÍA	80
ANEXOS	82
Anexo 1: Comparación de los modelos de Factoría de Software	82
Anexo 2: Niveles de la Estrategia	84
Anexo 3: Diferencias entre las metodologías ágiles y las tradicionales	84
Anexo 4: Estructura de la Norma ISO 9001: 2000	85
Anexo 5: Flujo general de trabajo de la línea de Software Educativo	85
Anexo 6: Diseño de la encuesta realizada a los integrantes del proyecto antes de probar la estrategia	86
Anexo 7: Diseño de la encuesta realizada a los integrantes del proyecto después de probar la estrategia	87
Anexo 8: Clasificación de una Factoría de Software	88
Anexo 9: Flujo de procesos de gestión del proyecto	88
Anexo 10: Datos del equipo de trabajo	89
Anexo 11: Mapa del proceso de elaboración	89
Anexo 12: Tabla descriptiva del flujo de procesos de elaboración	90
Anexo 13: Mapa de procesos de la prueba	92
Anexo 14: Tabla descriptiva del flujo de procesos de la prueba	92
Anexo 15: Mapa del proceso de la entrega	93
Anexo 16: Tabla descriptiva del flujo de procesos de la entrega	94
Anexo 17: Plantilla de solicitud de los cursos de capacitación	95
Anexo 18: Información del componente	95
Anexo 19: Estructura de organización de la información	95
Anexo 20: Conocimientos, habilidades y valores de los roles	96
Anexo 21: Diseño del Cuaderno del Ingeniero	98
Anexo 22: Diseño del Cuaderno de registro de tiempo	99
Anexo 23: Diseño del Resumen semanal de actividades	99
Anexo 24: Cuaderno de registro de defectos	100

Anexo 25: Descripción de las bases tecnológicas	100
Anexo 26: Políticas de Seguridad de la Información (PSI)	100
Anexo 27: Resultados de la encuesta inicial.....	101
Anexo 28: Resultados de la encuesta después de la estrategia.....	101
GLOSARIO DE TÉRMINOS Y SIGLAS.....	103

ÍNDICE DE FIGURAS

Figura 1.1 Modelo Basado en la Norma ISO 9001 y CMM	10
Figura 1.2 Modelo Eureka	11
Figura 1.3 Enfoque Software Bus	12
Figura 1.4 Modelo Clasificadorio	13
Figura 1.5 Modelo propuesto por Basili	14
Figura 1.6 Modelo Replicable.....	15
Figura 1.7 Modelo aplicando Inteligencia	16
Figura 1.8 Arquitectura de Transferencia de un modelo de Factoría de Software	24
Figura 2. 1 Estructura Organizacional	59

ÍNDICE DE TABLAS

Tabla 2.1 Instrumentación de la Estrategia.....	42
Tabla 2.2 Plan de riesgos, mitigación y contingencia para el proyecto	51
Tabla 2.3 Plan de riesgos, mitigación y contingencia para la implantación de la factoría	64

INTRODUCCIÓN

El propio desarrollo de la sociedad de la información impulsa el fortalecimiento de la investigación en muchos sectores de la sociedad. Especial importancia tiene la educación, por ser la base para la formación y preparación del personal necesario para la multiplicación de las capacidades productivas, responsables de todo desarrollo y renovación social.

En Cuba la industria del software está comenzando, con el paso de los años se ha ido reorganizando e industrializando sus procesos bajo nuevos paradigmas de calidad, por lo que la producción de software a gran escala presenta deficiencias que impiden el rápido avance que exige el mercado actual. Con el objetivo de fortalecer esta rama de la economía nacional surgió la Universidad de las Ciencias Informáticas (UCI), introduciendo un nuevo concepto basado en la vinculación estudio – trabajo donde la producción es un pilar fundamental y sus resultados contribuyen al desarrollo económico, político y social del país.

Para formar estudiantes con una alta preparación profesional, la universidad pone a su disposición una gran cantidad de recursos, tanto de información como tecnológicos, que ayudan en el desempeño de los proyectos productivos que desarrolla cada facultad de la universidad.

Los proyectos se realizan con diferentes entidades en el marco nacional e internacional y se distribuyen según las líneas de producción que tiene establecida la universidad. Entre los proyectos que pertenecen a la línea de producción de software educativo están los dirigidos a elaborar software educativo para Venezuela como parte del Convenio Integral Cuba – Venezuela.

A raíz de esto se formó el proyecto CNTI – Contenidos Educativos, con el objetivo de desarrollar gran cantidad de productos educativos multimedia orientados a la Web. Esta elevada demanda de software para la educación crece cada día a medida que se expanden las Tecnologías de la Información y las Comunicaciones (TICs), por lo que se deben trazar estrategias que, a corto y mediano plazo, proporcionen un incremento de la eficiencia, flexibilidad y adaptabilidad a los cambios, calidad del proceso de desarrollo del software y del producto, orientado a la mejora continua de los procesos, cumplimiento de compromisos, comunicación y planificación.

Para constatar el problema que da origen a la investigación y cumplir el objetivo propuesto se utiliza la metodología de investigación descriptiva y se desarrollará para el proyecto CNTI – Contenidos Educativos con la ayuda de diferentes técnicas.

Como parte de la investigación se realizaron encuestas para valorar el estado y funcionamiento del proyecto, lo que demostró que existía inexperiencia en la dirección, gestión de las actividades dentro del proyecto y desconocimiento de las herramientas tecnológicas para desarrollar los productos, provocando afectaciones durante el proceso de desarrollo. El líder de proyecto no tenía correctamente definidos los roles y las responsabilidades por cada flujo de trabajo, lo que trajo consigo una sobrecarga de responsabilidad en algunos de los trabajadores y esto les impedía dedicarle la atención requerida a cada tarea. Además, no se contaba con una planificación de las actividades personales ni en equipos que realizara estimaciones basadas en el conocimiento real y en la capacidad productiva. El proceso no estaba guiado por estándares internacionales, afectándose la efectividad de los equipos de desarrollo. Otro de los problemas encontrados durante la investigación fue relacionado con el diseño gráfico, ya que a veces no se parecían al guión entregado por el cliente, provocando que el desarrollador no comprendiera el diseño cuando iba a integrar los contenidos. Todo esto unido a la falta de dominio de las tecnologías, lograron que cada miembro del proyecto se sintiera desorientado y sin saber qué hacer en cada momento ni a quién dirigirse. Por último se reutilizaban pocos componentes y estos no se documentaban para facilitar su búsqueda y utilización en productos posteriores.

La investigación en busca de la mejora en los procesos de desarrollo de software recoge diversas alternativas, entre ellas se encuentran las Factorías de Software que tienen como propósito disminuir el tiempo de respuesta a la demanda de producción de software mediante la reutilización de códigos probados anteriormente, guiar el proceso de desarrollo de software mediante estándares de calidad para la mejora continua de procesos en todas sus áreas y organizar la estructura de trabajo mediante una correcta gestión de proyectos. Para lograr esto se necesitan definir estrategias que transfieran este nuevo enfoque a la práctica laboral. Prever una secuencia de pasos que guíen el proceso desde el plano ideal a la realidad permite que se cumpla el objetivo y que se obtengan los resultados esperados.

Los aportes prácticos que se esperan son principalmente los de contribuir a la mejora y organización del proceso de desarrollo de software para multimedia con tecnología Web y servir de referencia para proyectos similares. Apoyar la especialización de los trabajadores por roles, según el área de trabajo que prefieran, siguiendo normas de calidad que conlleven a la eliminación de riesgos por desconocimiento de las responsabilidades o por un trabajo inadecuado. El enfoque de Factoría de Software se apoya en la reutilización de componentes, lo que provoca una disminución en el tiempo de vida del proyecto. Por lo tanto, se piensa que al poner en práctica una estrategia que transfiera este enfoque se disminuya el costo de fabricación de los productos.

El **problema científico** de este trabajo investigativo trazado como resultado de un análisis detallado de la situación problemática actual y de la búsqueda de información a partir de la misma consiste en *¿Cómo las deficiencias en el modelo de producción de los contenidos educativos afectan la producción?*

Por tanto el **objeto de investigación** es el *“proceso de desarrollo de software educativo en la UCI”*.

El **objetivo** será: *“Diseñar una estrategia de transferencia de un modelo de Factoría de Software para el desarrollo de Contenidos Educativos”*.

De ello se deriva que el **campo de acción** es el *“proceso de elaboración de contenidos educativos”*.

La **hipótesis** formulada basada en el problema y en el objetivo propuesto como meta plantea que *“si se estudia el proceso de producción de software existente en el proyecto y se elabora una estrategia de transferencia de un modelo de Factoría de Software se podrían eliminar las deficiencias existentes”*.

Para cumplir con el objetivo planteado y resolver los problemas mencionados se proponen las siguientes **tareas**:

1. Caracterizar y definir el enfoque de Factoría de Software y los modelos que se han aplicado en el mundo.
2. Caracterizar y definir el proceso de elaboración de software educativo y el funcionamiento del proyecto CNTI – Contenidos Educativos.
3. Diseñar y probar algunos de los elementos de la estrategia de transferencia del modelo de Factoría de Software.

Esta propuesta está basada en la información obtenida durante la investigación y en la experiencia adquirida por países y personas que han decidido utilizar el enfoque de Factoría de Software, para de una forma u otra contribuir a fomentar esta última tendencia dentro del campo de la informática.

La estructura del trabajo se muestra a continuación, acompañada de una breve explicación por cada capítulo.

Capítulo 1: Fundamentación Teórica. Aborda conceptos relacionados con el tema de Factoría de Software. Se mencionan los modelos de Factorías de Software más representativos a nivel internacional, las características esenciales de cada uno y su experiencia vista desde las empresas en donde se aplicó este enfoque. Se abordan aspectos sobre la definición de estrategias.

Capítulo 2: Estrategia de Transferencia. Describe los métodos, procedimientos y técnicas utilizadas para llevar a cabo la investigación científica y propone la estrategia de transferencia del modelo seleccionado.

Capítulo 3: Resultados. Muestra los resultados obtenidos de las encuestas y entrevistas aplicadas, así como los resultados de probar algunos de los elementos de la estrategia propuesta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se abordarán un grupo de conceptos importantes relacionados con la obtención de una estrategia de transferencia de una Factoría de Software para la línea de contenidos educativos. Se describen los modelos más representativos encontrados durante la revisión bibliográfica, los principales estándares de calidad para el enfoque, los resultados de algunas empresas que lo adoptaron y las metodologías para el desarrollo de software educativo. También se presentarán definiciones de estrategia, los enfoques y niveles que las caracterizan, el proceso de transferencia; así como las diferentes tecnologías a utilizar en la Factoría de Software.

1.2 Factoría de Software

Según Ivan Aaen, “la fábrica es una organización habitada por personas comprometidas en un esfuerzo común, el trabajo es organizado de una manera o de otra, se usa la estandarización para la coordinación y formalización, y la sistematización es importante.”(IVAN AAEN 1997)

El proceso fabril tradicional utiliza la técnica de línea de producción y montaje con el objetivo de reducir los costos de producción por la vía de la estandarización de productos y procesos. La producción por montaje se caracteriza por encadenar secuencias de procesos que convergen hacia una línea continua en la que se ensamblan los productos finales.

Entre las industrias que trabajan por montaje se cuentan algunas de las actividades productivas de mayor relevancia para la economía actual, principalmente las mecánicas: automóviles, motores, tractores, electrodomésticos, electrónicos, etc.

Arnoldo Díaz, director general de Certum plantea que “el fundamento para establecer fábricas de software, fue basado en tratar de obtener los beneficios que las líneas de producción industrial produjeron en la calidad de los productos así como la productividad lograda.” (ARNOLDO DÍAZ OLAVARRIETA 2006)

Por lo tanto, es un pilar fundamental para su éxito que las Factorías de Software se nutran de esta experiencia, aplicándola a su producción a través de la especialización del trabajo por líneas de producción, donde cada una tenga un proceso y un flujo de trabajo correctamente definidos.

1.2.1 Definiciones de Factoría de Software

A fines de los años 60's surge en la industria del software el concepto de Fábrica de Software y ha evolucionado a medida que se han definido técnicas y métodos, además de la aparición de diversas herramientas de software. Aunque el término Factoría de Software no es nuevo, desde hace varias décadas ha adquirido una especial relevancia con el desarrollo de modelos. A continuación se enuncian varios conceptos de Factoría de Software dados por distintos autores:

Cusumano plantea en 1989 que "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." (CUSUMANO 1989)

Cantone enuncia en una conferencia pronunciada en 1992 que "Una Factoría de Software debe, para ser flexible, ser capaz de producir varios tipos de productos; llevar a cabo conceptos de Ingeniería de Software (metodología, herramientas, gestión de la configuración), y también ser capaz de estudiar, diseñar, implementar y mejorar sus sistemas y procesos." (CANTONE 1992)

Fernandes y Texeira definen en el 2004 que "Una Factoría de Software es una organización con procesos estructurados, controlados y mejorados de forma continua, considerando principios de Ingeniería Industrial, orientados a dar respuesta a múltiples demandas de distintas naturaleza y alcance. Dirigida a la creación de productos de software, conforme a los requerimientos documentados de los usuarios y clientes, de la forma más productiva y económica posible" (FERNANDES 2004)

EN 1992 Ferneston afirma "Una organización fabril para el desarrollo de software debe tener claro el asunto del "software único", es decir, todo software es único, pero algunas partes de ellos se pueden repetir en varios proyectos. El proceso industrial debe contener el desarrollo, almacenamiento y montaje de partes reutilizables en un producto" (FERNSTROM 1992)

Basili en 1992 afirma que "Una organización con características de Factoría de Software debe poseer una estructura de construcción de software basada en componentes. Los componentes utilizados en la construcción del software pueden ser desarrollados por una unidad de producción de componentes (factoría de componentes). La factoría de componentes es la base para la implementación de una Factoría de Software." (BASILI 1992)

Para reafirmar lo antes mencionado dice que "Una Factoría de Software 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.” (BASILI 1992)

Por su parte Yaimí Trujillo Casañola en el evento Informática 2007 afirma que una Factoría de Software es “una organización estructurada creada para el desarrollo de software, con procesos estandarizados, repetibles, gerenciales y principalmente mejorable continuamente.” (CASAÑOLA 2007)

Estos autores concuerdan en que una Factoría de Software debe estar guiada por metodologías de desarrollo de software contempladas en la disciplina de la Ingeniería de Software y orientada a la mejora continua de sus procesos. Debe definir los roles y tareas que permitan producir software en grandes cantidades. Este aumento en la productividad debe estar apoyado por la reutilización de muchos de los componentes que integran un software y del conocimiento adquirido, con el fin de elevar la calidad, disminuir los riesgos, aumentar la productividad y minimizar el tiempo de desarrollo.

1.2.2 Aspectos importantes sobre las Factorías de Software

Yanosky Ríos en su trabajo de investigación recoge una serie de características inherentes a una estructura fabril para software y que concuerda con las definiciones de una Factoría de Software:

- “Proceso definido y estandarizado para el desarrollo de software.
- Interacción controlada con el cliente, alto poder de entendimiento.
- Estimación de costos y tiempo basados en el conocimiento real de la capacidad productiva, mediante métodos de obtención mediante datos históricos.
- Control de los recursos humanos involucrados.
- Planificación y control de la producción
- Control y almacenamiento en bibliotecas de componentes de software (documentos, código, métodos, etc.)
- Control del estado de ejecución de todas las demandas.
- La producción de software debe estar fuertemente basada en métodos y técnicas estandarizadas.
- Capacitación de los recursos humanos.
- Garantía de la calidad del producto.
- Mecanismos de control de costos.
- Mejora continua del proceso.” (YANOSKY RÍOS 2005)

La función principal de una Factoría de Software es definir un proceso estándar para el desarrollo de software, donde exista una fuerte comunicación con el cliente y las estimaciones se realicen basadas en los datos históricos como fruto de experiencias anteriores. En una fábrica de este tipo la producción tiene que estar bien controlada, así como los recursos humanos con que cuenta para asumir las tareas, las que pueden desarrollarse en menor tiempo si se cuenta con una biblioteca de componentes que los mantenga actualizados y correctamente documentados para su reutilización. Reutilizar un componente garantiza que el producto en elaboración cuente con la misma calidad que su antecesor.

Por tanto, se puede afirmar que una Factoría de Software es una organización dedicada a la producción de software con procesos industrializados, donde se comparten recursos y conocimientos según su línea de producción. Estas líneas agrupan los proyectos que trabajan sobre una misma plataforma tecnológica que posibilitan crear un esquema para el desarrollo y mantenimiento de sistemas similares.

Yanosky Ríos también aborda los principales objetivos de una Factoría de Software que permiten industrializar su funcionamiento. Estos objetivos son los siguientes:

- 1- “Industrializar el desarrollo de sistemas de software.
- 2- Producción de software a gran escala.
- 3- Lograr una alta productividad en el desarrollo de software.
- 4- Establecer una línea de producción.
- 5- Mejora continua de los procesos.
- 6- Estimación de costos y plazos extremadamente precisos.
- 7- Reducción de los costos de producción.
- 8- Lograr una buena gestión de la calidad.
- 9- Especializar al profesional en una tarea específica del proceso, concentrando sus esfuerzos en dicha tarea.” (YANOSKY RÍOS 2005)

Estos objetivos dentro del enfoque de Factoría de Software se consiguen sobre la base de principios básicos, los cuales son aplicados en empresas como criterios fundamentales para fabricar software, como la compañía Ibermática, con más de 30 años de experiencia y que se ha consolidado como una de las cinco principales compañías de servicios en el sector de las TIC en España. Sus centros de producción de soluciones y servicios aplican de forma sistemática conceptos avanzados en materia de estandarización e industrialización del software. Los principios que se muestran a continuación pertenecen a esta importante empresa:

Estandarización y normalización: Para diseñar y crear componentes que permitan fabricar el propio software, es necesario primeramente conocer qué software se desea fabricar. “En la Factoría de Software es necesario concretar el código que se va a desarrollar para poder contar con el conjunto de componentes necesarios. Es imprescindible, por lo tanto, estandarizar y normalizar los resultados (programas) para diseñar y crear los componentes necesarios para obtenerlos.” (IBERMÁTICA 2007) Para lograr esto, es importante contar con el apoyo de los mejores expertos de la organización ya que toda esa experiencia y conocimiento recogidos en los componentes, permiten obtener eficaces programas estandarizados con garantía de calidad.

Componentes basados en similitud: Durante el desarrollo de software de una compañía se producen siempre programas distintos, mientras que en una factoría industrial se fabrican grandes tiradas del mismo producto. Por lo tanto, “el diseño y la fabricación de los componentes de la factoría se realiza sobre la base de este concepto de similitud, lo que permite al equipo de desarrollo trabajar con un número muy reducido de componentes flexibles que adaptan su uso en la fabricación de múltiples programas y al mismo tiempo, alcanzan altas cotas de reutilización de código estándar y normalizado.” (IBERMÁTICA 2007)

Reutilización global: En la factoría, “el concepto de reutilización global de código se refiere a todo el equipo de desarrollo y no exclusivamente al concepto de reutilización individual en el plano del programador.” (IBERMÁTICA 2007) La existencia en ésta de un conjunto de componentes estándar para todos los programas y todos los programadores hace que la capacidad de reutilización en la fabricación de programas no dependa de cada individuo, sino que sea una característica propia de la factoría y esté gestionada desde un único lugar.

Soporte a los estándares de cada organización: “En toda organización con una cierta experiencia en desarrollo de software existe un conjunto de reglas, usos y estándares sobre desarrollo y codificación del software. Con la implantación de una Factoría de Software, estas reglas, salvo el caso de que fueran contradictorias con alguno de los conceptos fundamentales de la factoría, se deben incorporar.” (IBERMÁTICA 2007) Así se asegura que la factoría produzca el software de manera que el código de los programas sea exactamente el que ha definido; es decir, el que ha definido cada compañía, no el que venga determinado por el uso de herramientas concretas.

1.3 Modelos de Factoría de Software

En este epígrafe se hace referencia a los modelos de Factoría de Software más representativos encontrados durante la investigación y que se encuentran respaldados por la experiencia de empresas y entidades que los han adaptado. Los modelos son los siguientes:

- Modelo basado en la norma ISO 9001 y CMM.
- Modelo Eureka.
- Modelo Clasificadorio.
- Modelo propuesto por Basili.
- Modelo Replicable.
- Modelo aplicando Inteligencia

1.3.1 Modelo basado en la norma ISO 9001 y CMM

En este modelo se hace una división de los elementos fundamentales de una Factoría de Software en cinco entidades: Técnicas, Proceso, Trabajadores involucrados, Gestión de la factoría y Activos del proceso, herramientas y componentes de código. En la figura 1.1 se puede observar cada una de las entidades mencionadas y las relaciones entre ellas.

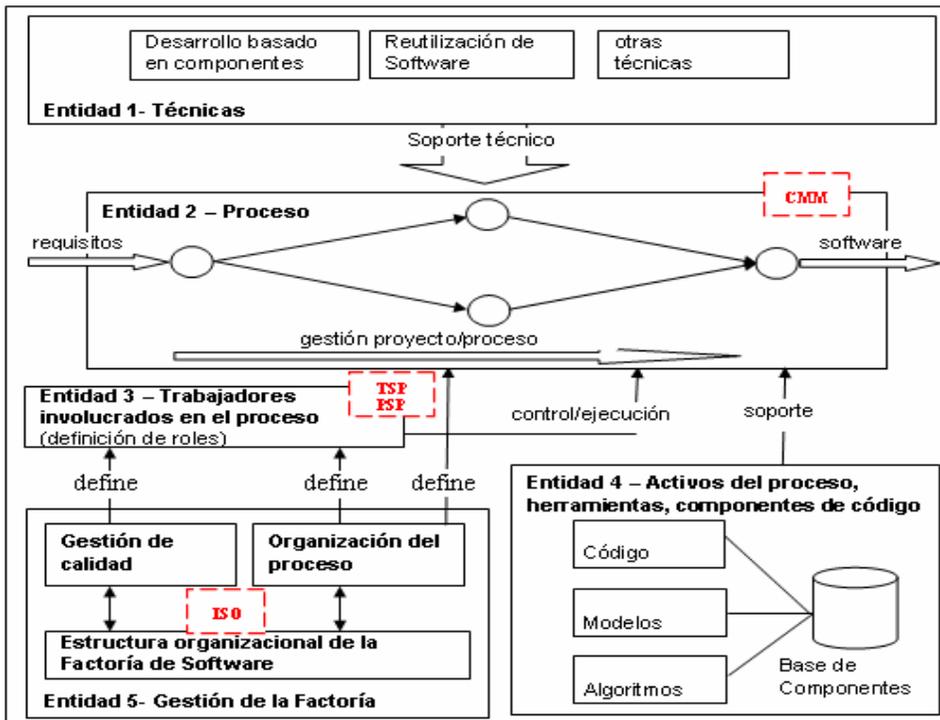


Figura 1.1 Modelo Basado en la Norma ISO 9001 y CMM

Yanosky explica estas entidades y relaciones en su trabajo de la siguiente forma: “La entidad Técnicas provee el soporte técnico y conceptual para la definición del proceso. Este es guiado por el estándar de calidad CMM, los requisitos de calidad para la organización de la factoría son definidos por la norma ISO 9001. El modelo toma la norma ISO 9001 como un estándar utilizado en el contexto industrial cuyo enfoque está en el sistema de calidad organizacional, propone un conjunto de principios probados para mejorar la calidad final del producto mediante mejoras en la organización de la empresa. CMM es designado para la industria del software, de este modo las áreas claves proveen detalles importantes para la evaluación y mejora del proceso de desarrollo, su propósito es guiar a las organizaciones en la selección de estrategias de mejora determinando la madurez del proceso actual e identificando los puntos importantes que se deben estudiar y trabajar para mejorar tanto el proceso como la calidad del software. La entidad Gestión de la Factoría define, a través de las sub-entidades Gestión de Calidad y Organización del Modelo de Proceso, los trabajadores involucrados en el proceso de desarrollo de software y sus roles. La sub-entidad Organización del Modelo de Proceso define características y organización del proceso de desarrollo de software. Los Trabajadores Involucrados son guiados por los modelos PSP (Personal

Software Process) y TSP (Team Software Process). Los activos del proceso, las herramientas y los componentes de código dan soporte al proceso de desarrollo de software.” (YANOSKY RÍOS 2005)

Este modelo es relevante principalmente por los estándares de calidad que definen su nombre, el uso de la norma ISO 9001 y de CMM garantizan que los procesos se desarrollen con la calidad necesaria. El enfoque principal de este modelo está en el proceso del software en función de la mejora continua para lograr una buena estimación de los costos y tiempo. Otro elemento importante es que engloba las actividades en entidades, pero no las especifica en su totalidad por lo que queda muy general.

1.3.2 Modelo Eureka

El modelo de factoría Eureka surge como iniciativa de la comunidad europea para ayudar a la industria a mejorar las tecnologías de fabricación y fortalecer su competitividad en el mercado; además, forma parte del proyecto Eureka para la investigación y desarrollo (I+D). El modelo fabril está compuesto de proceso, reglas, herramientas, información, trabajadores y equipamiento (Figura 1.2).

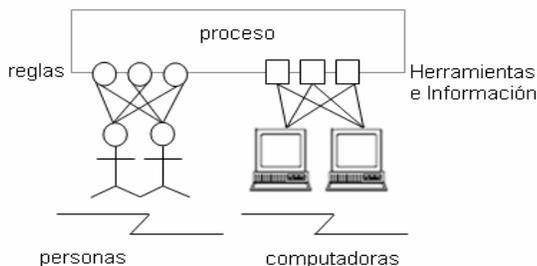


Figura 1.2 Modelo Eureka

“El proceso de desarrollo está compuesto por reglas, las que son definidas por las personas involucradas en el ambiente de desarrollo de software y constituyen patrones a seguir, algoritmos, métodos de desarrollo de software. Las herramientas e información almacenada, soportan la automatización del proceso de desarrollo. El modelo posee características giradas al proceso de desarrollo de software distribuido, en el mismo se sigue el enfoque software bus. Enfoque que estipula reglas de conexión de componentes en la construcción de un software” (YANOSKY RÍOS 2005)

El enfoque software bus puede observarse en la siguiente figura:

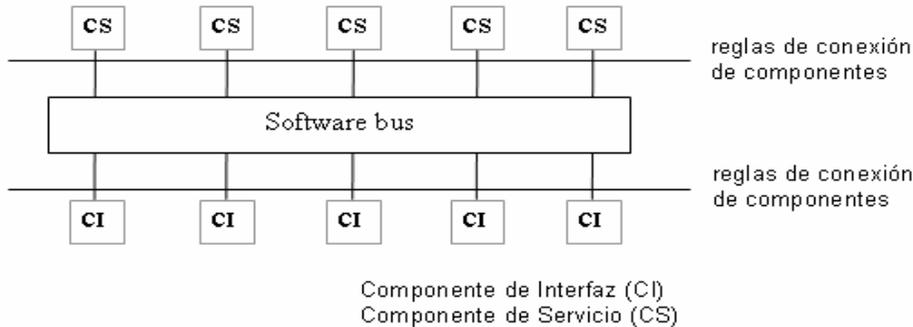


Figura 1.3 Enfoque Software Bus

Según Yaimí, “el modelo da una visión de cómo se puede desarrollar un producto en partes manejables y después la unión de estas para formar el producto final.”(CASAÑOLA 2007)

La importancia de este modelo radica en el desarrollo de software distribuido, porque elimina las limitaciones geográficas, cada Factoría de Software puede trabajar por separado, guiándose por las reglas establecidas para la integración de los componentes desarrollados por cada una de ellas. Establece infraestructuras tecnológicas y se apoya en herramientas automatizadas para la integración de los componentes. Un aspecto negativo es que no abarca temas como la gestión de proyectos, recursos, metodologías de desarrollo o el uso de estándares de calidad.

1.3.3 Modelo Clasificadorio

El Modelo Clasificadorio divide las factorías de acuerdo al alcance dentro del proceso de desarrollo de software y las clasifica en: Factoría de Proyectos Ampliada, Factoría de Proyectos de Software, Factoría de Proyectos Físicos, Factoría de Programas. Estas clasificaciones se pueden observar en la figura 1.4.

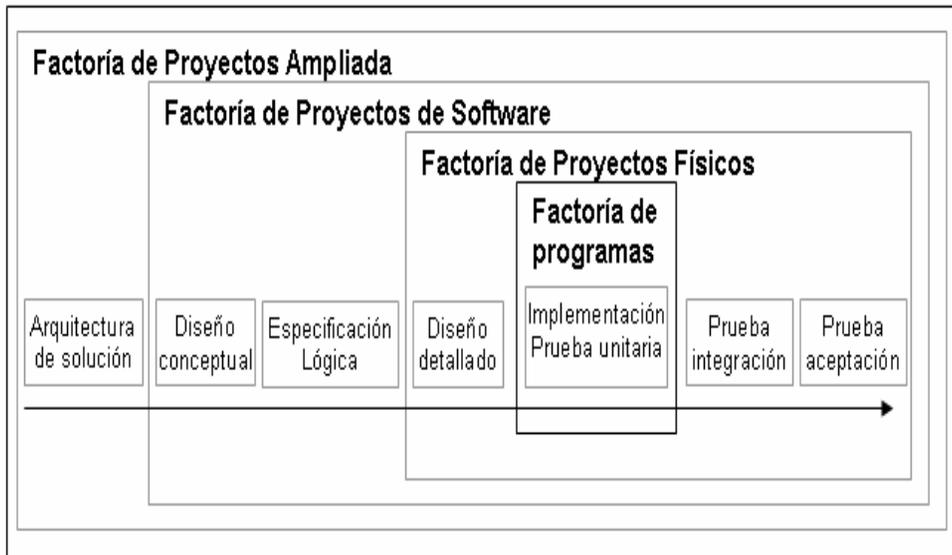


Figura 1.4 Modelo Clasificador

“Una *Factoría de Proyectos Ampliada* comprende el concepto de arquitectura de solución. La arquitectura de solución es una etapa anterior al diseño conceptual del software, la cual se ocupa en proyectar una solución en la que el software está formado por los componentes más significativos arquitectónicamente, se definen los principios que orientan el diseño y evolución del software. La arquitectura de solución puede contener, además del software, definición de procesos, definición de equipamiento, infraestructura de redes, plataforma de desarrollo, patrones a seguir.

La *Factoría de Proyectos de Software* abarca todo el ciclo de vida sistémico para la realización del software, correspondiente al análisis, diseño, implementación, prueba e implantación. En este tipo de factoías se tiene un conocimiento al detalle del negocio a automatizar.

La *Factoría de Proyectos Físicos* se abstrae del enfoque sistémico del software, se dedica al diseño, implementación y prueba. No se tiene un pleno conocimiento del negocio.

La *Factoría de Programas*, considerada la menor de las entidades, tiene como objetivos desarrollar componentes de código para la construcción del software. Esta factoía no se preocupa del contexto sistémico ni del diseño, se ocupa de producir código según las especificaciones del diseño. Posee como entrada la especificación del diseño de una parte del software y su salida es un componente de código que formará parte del software a desarrollar.” (FERNANDES 2004)

El aporte principal del modelo es la clasificación de las Factorías de Software de acuerdo a los flujos de trabajo dentro del proceso de desarrollo de software. Sin embargo, solo se limita a clasificar su alcance y deja sin definir áreas claves como la gestión del proyecto y soporte.

1.3.4 Modelo propuesto por Basili

Este modelo divide una Factoría de Software en dos grandes entidades. “El modelo se divide en organización basada en proyectos de software (unidad de producción de software), y factoría de componentes (unidad de producción de componentes). La organización basada en proyectos realiza las solicitudes de productos (componentes para la construcción del software), de datos (estadística para la estimación de costo y plazos) y de planos (modelos, métodos para el análisis y diseño de software) a la factoría de componentes. La factoría de componentes posee una base de componentes reutilizables, de la cual se apoya para dar respuesta a las solicitudes hechas por la unidad de producción de software. En respuesta a la solicitud la organización basada en proyectos recibe los modelos y componentes para la construcción del software, además de estadísticas y datos históricos que se encuentran en la base de componentes.” (YANOSKY RÍOS 2005). El modelo puede verse en la figura 1.5.

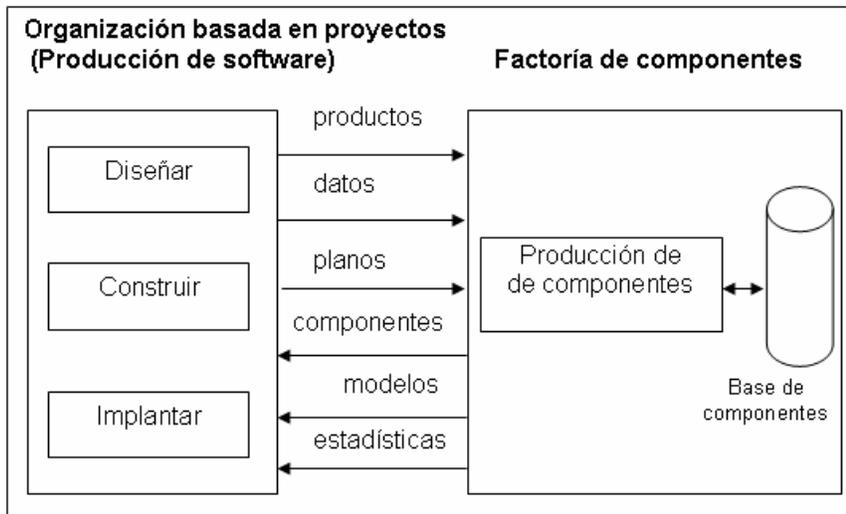


Figura 1.5 Modelo propuesto por Basili

El rasgo más distintivo del modelo propuesto por Basili es la división en unidades, principio básico para aumentar la productividad; así cada unidad se especializa en determinadas actividades, ganando en experiencia, lo que apoyado con la reutilización de componentes, permite ganar en tiempo de desarrollo y

en la efectividad de los productos desarrollados. El modelo se encuentra incompleto porque su enfoque principal está en el proceso de desarrollo del software, dejando sin definir el resto de las áreas involucradas en la producción.

1.3.5 Modelo Replicable

El presente modelo reúne las características más importantes de los modelos anteriores. Plantea que una Factoría de Software debe poseer un modelo de organización de la producción, una unidad de producción de componentes y una unidad de producción de software, ambas deben poseer un proceso guiado por un modelo de calidad de software. Además describe las técnicas, herramientas, roles y actividades dentro del proceso. En la figura 1.6 se encuentran las relaciones entre los conceptos.

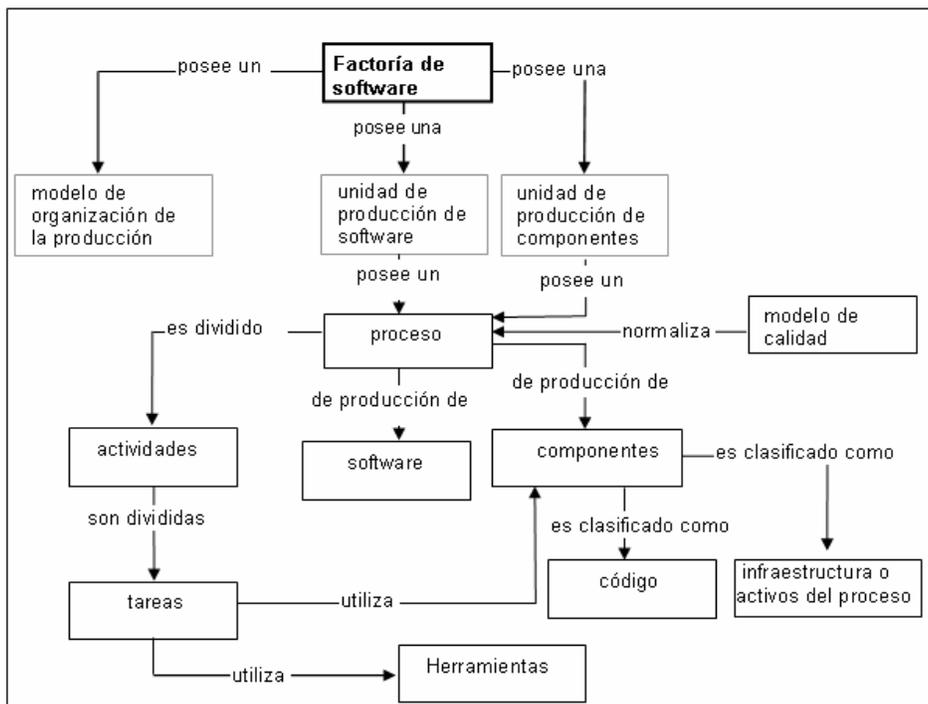


Figura 1.6 Modelo Replicable

La organización de la producción se divide en cinco áreas y define las actividades para cada una de ellas, “donde la unión de las áreas de análisis de sistemas, diseño de software y construcción de software forman el ámbito de negocio del modelo de producción. El ámbito de negocio incluye la interacción entre el cliente y la Fábrica de Software.

Las áreas de producción de componentes de infraestructura y componentes de código forman el ámbito interno del modelo, el que es transparente a los ojos del cliente de la factoría. Este ámbito es el

responsable de los subproductos creados, componentes para la construcción del sistema.” (YANOSKY RÍOS 2005)

Este modelo es uno de los que mejor definen el proceso de producción, está muy detallado por cada actividad pero no rige el proceso por estándares de calidad ni por metodologías de ingeniería de software reconocidas, además de que tampoco abarca completamente el área referida a la gestión de proyecto.

1.3.6 Modelo aplicando Inteligencia

Este modelo de Factoría de Software fue propuesto por Yaimí Trujillo Casañola para la Universidad de las Ciencias Informáticas. El mismo define entidades, sus relaciones y los objetivos de cada una de ellas. Tiene en cuenta estándares de calidad reconocidos, mecanismos para apoyar el proceso de desarrollo y la organización. Define roles, responsabilidades y procesos según la metodología utilizada. Las entidades son: Bases tecnológicas, Proceso, Personas, Repositorio de componentes, Gestión de Proyectos e Inteligencia y sus relaciones se pueden apreciar en la figura 1.7.

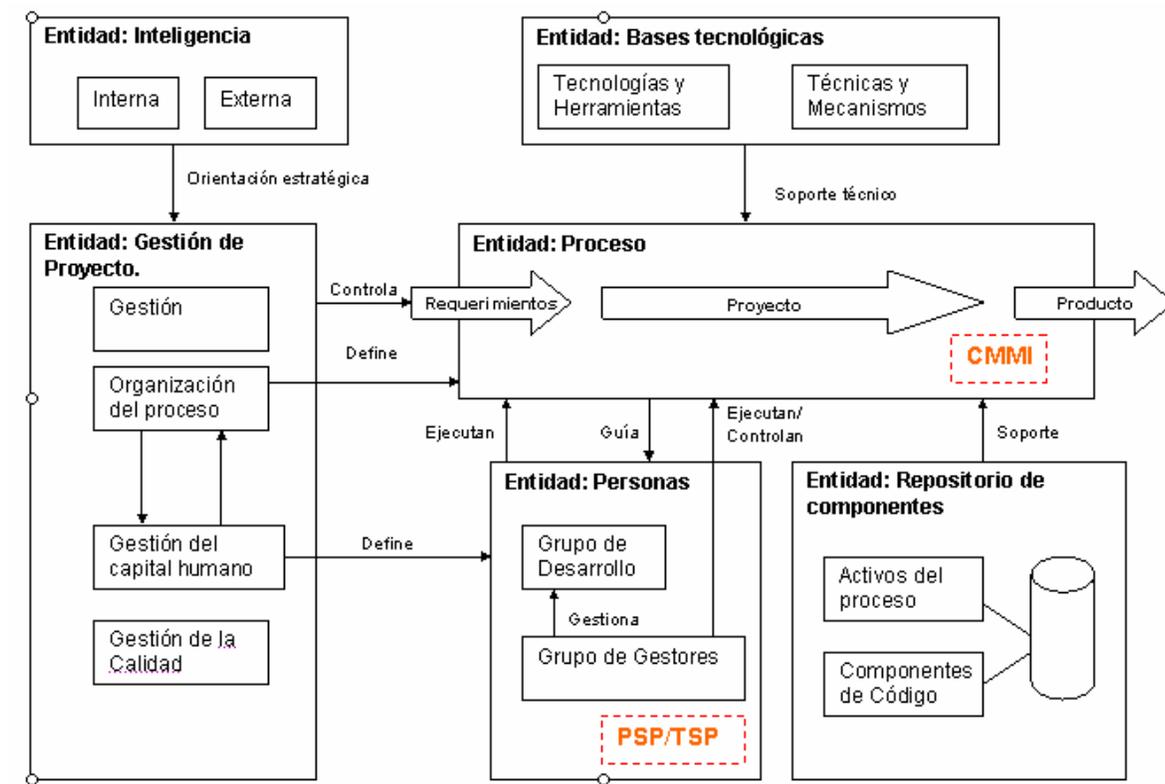


Figura 1.7 Modelo aplicando Inteligencia

“La descripción de la estructura y composición del modelo se basa en que el resultado final de un proyecto en la Factoría de Software es un producto, que toma forma durante su desarrollo gracias a los desarrolladores representados en la entidad Personas. Estas se representan de dos formas, mediante la sub-entidad Grupo de desarrollo y Gestores de la Factoría. El equipo de desarrollo lo forman las personas involucradas directamente en el proceso, el de gestores comprende el equipo de dirección de la misma, encargados del control y gestión del grupo de desarrollo, los cuales ejecutan las actividades o flujos de trabajo. A su vez son guiados por el proceso de desarrollo de software, representado en el modelo mediante la entidad Proceso de desarrollo.

El proceso es automatizado y soportado por diversas tecnologías y herramientas, representadas en la entidad Técnicas y Herramientas. La reutilización tiene efectos muy positivos en el desarrollo de software, entre estos efectos están el aumento en la productividad y calidad así como la reducción del tiempo de desarrollo. Para dar soporte al proceso en este sentido la factoría cuenta con una base de componentes reutilizables, representada en la entidad Repositorio de Componentes. Todo esto es gestionado desde la entidad Gestión de la Factoría la cual recibe la orientación estratégica de la entidad Centro de Inteligencia.” (CASAÑOLA 2007)

Este modelo recoge las características más trascendentales de los modelos vistos anteriormente, utiliza estándares internacionales, define el proceso por metodologías de desarrollo y apoya la reutilización. Además de definir el proceso, le brinda gran importancia a las áreas de gestión de proyecto, organización y calidad del proceso de desarrollo del software. También busca la manera de gestionar el conocimiento, la experiencia de la empresa y las características del entorno con la introducción de la entidad Centro de inteligencia.

1.3.7 Análisis comparativo de los modelos

El análisis de los modelos de Factoría de Software vistos anteriormente, según las características especificadas para una Factoría de Software, permitió obtener elementos para seleccionar el modelo para el cual se le diseñará la estrategia. En el anexo 1 aparece una tabla comparativa que resume las características de estos modelos.

El modelo basado en la norma ISO 9001 y CMM se puede adaptar según las necesidades de la factoría. Refleja la necesidad de la reutilización de componentes y agrupa sus actividades por áreas. El modelo utiliza estándares para la calidad y la mejora continua del proceso.

El modelo Eureka al igual que el anterior, guía su proceso de desarrollo por reglas y tiene como característica principal que se enfoca a la producción de software de forma distribuida o por componentes. El modelo Clasificador y el propuesto por Basili se caracterizan por buscar la especialización de las factorías. El primero las define según el alcance del proceso de desarrollo mientras que Basili agrupa las actividades de una factoría en dos unidades, buscando la especialización y el aumento de la reutilización de los componentes producidos.

Los restantes dos modelos, el Replicable y el aplicando Inteligencia, tienen como característica fundamental que se nutren de los elementos más sobresalientes de los anteriores, definen con mayor precisión las actividades y tareas a realizar dentro de cada flujo de trabajo, aunque solo el modelo de Factoría de Software aplicando Inteligencia se dedica a definir el área de gestión del proyecto, aspecto esencial que determina el éxito de un proyecto de software. Además, ninguno de los otros modelos incluye el tratamiento de la orientación estratégica para gestionar el conocimiento generado.

1.4 Experiencia Internacional

Como parte de la investigación se han encontrado varias organizaciones que ponen en práctica el enfoque de Factoría de Software. Estas empresas se ubican en varios países y han obtenido una serie de resultados que les ha reportado grandes beneficios en todos sus sectores.

Ivan Aaen afirma que “la primera compañía en el mundo que nombró una organización del software como una factoría fue Hitachi en 1969, la segunda fue implantada entre los años 1975-1976 bajo el nombre de Corporación de Desarrollo de Sistemas, por un líder americano en el campo del software personalizado.” (IVAN AAEN 1997)

En la década de los '70 se destacaron algunas empresas japonesas que adoptaron trabajar con Factorías de Software, entre ellas están Toshiba y Fujitsu.

Otro ejemplo es AZERTIA, un Grupo multinacional español de empresas de Tecnologías de la Información que pertenece a la Corporación IBV (Instituto de Biomecánica de Valencia). Tiene más de 3.300 profesionales que se distribuyen por nueve países (España, Portugal, EE.UU. Argentina, Brasil, Colombia, México, Puerto Rico y Venezuela). Investiga e innova mediante alianzas y acuerdos estratégicos con los principales proveedores de tecnologías líderes en el mercado, que le permiten complementar sus soluciones y adaptarlas a las necesidades concretas de cada cliente. Surgió en el 2001 como la integración de Centrisa, Keon y TeleInformática, dependencias de IBV también. (AZERTIA 2001)

Ibermática es una de las principales compañías de servicios en Tecnologías de la Información en el mercado español. Su actividad se centra en las siguientes áreas: consultoría TIC, equipamientos e infraestructuras, integración de sistemas de información, outsourcing e implantación de soluciones integradas de gestión empresarial para pequeñas y medianas empresas (PYME). En sus más de 30 años de experiencia en el sector de las TIC, Ibermática se ha distinguido por la búsqueda permanente de la excelencia y una apuesta decidida por la calidad y la innovación. Surgió en 1973 para brindar servicios informáticos al mercado español y desde entonces evoluciona constantemente para adaptarse a los cambios del mercado informático mundial. Por esto, en el 2005 Ibermática creó El Instituto Ibermática de Innovación, un centro de investigación que dirige la aplicación de modelos de innovación tanto en la propia Ibermática como en otras organizaciones, a través de la implantación de soluciones y la creación de una metodología para medir la innovación. Este instituto basó su funcionamiento en el enfoque de Factoría de Software, aplicando de forma sistemática los conceptos más avanzados en materia de estandarización e industrialización del software, permitiéndole a Ibermática ganar mayor prestigio a nivel nacional e internacional. (IBERMÁTICA 2007)

Vector Software Factory es otra de las empresas que han aplicado el enfoque de Factoría de Software como solución para sus procesos. Esta red de producción de soluciones de software se basa en el desarrollo de componentes y construcción de componentes de software avanzados, basados en componentes reutilizables. Surgió en el 2000 con el objetivo de desarrollar una Red de Centros de Producción de Componentes Software orientada al desarrollo y mantenimiento de componentes y soluciones complejas para empresas y organizaciones de diversos sectores. Centra su atención en el cliente por lo que tiene un equipo de consultoría directamente donde éste radica y dispone de centros de producción remotos con la infraestructura necesaria para el desarrollo de las aplicaciones. (FACTORY 2000)

Desde la implantación de la primera Factoría de Software Hitachi hasta la fecha se puede observar que las empresas que han adoptado este enfoque han obtenido importantes beneficios, sobre todo en el cumplimiento de los costos y tiempos estimados, la calidad de los productos y la satisfacción de sus clientes. Estas empresas cuentan con equipos especializados y repositorios de componentes reutilizables.

1.5 Estrategia

El término de estrategia es tan antiguo como la propia historia de las guerras militares. Según la bibliografía consultada, el término tiene un origen griego y se utilizaba para nombrar a una o varias personas con la capacidad para dirigir batallas o negociar la culminación de las mismas. A la persona encargada de crear la estrategia se le conocía como “estratega” y tenía el objetivo de definir el camino para alcanzar la victoria.

El ingeniero Serguei Alejandro Martin en su investigación sobre planeación estratégica apoya esta reseña histórica, “la estrategia de un ejercito también podría definirse como el patrón de acciones que se realizan para poder responder al enemigo. Los generales no solamente tenían que planear, sino también actuar.”(MARTIN 2002)

En este epígrafe se tratarán temas importantes relacionados con la estrategia de transferencia propuesta para el trabajo.

1.5.1 Definiciones de Estrategia

El diccionario de la lengua española brinda dos definiciones de estrategia. La primera es consecuente con el surgimiento del término 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)

La definición que brinda el Diccionario Pequeño Larousse de 1992 recoge como concepto de estrategia el “arte de dirigir las operaciones, habilidad para dirigir un asunto”(GARCÍA PELAYO Y GROSS 1992)

Mintzberg define además que una estrategia es un "Proceso racional a través del cual el estratega se abstrae del pasado para situarse mentalmente en un estado futuro deseado y desde esa posición tomar todas las decisiones necesarias en el presente para alcanzar dicho estado." (MINTZBERG H. Y QUINN 1993) El autor también se refiere al término estrategia como “Un curso de acciones conscientemente deseado y determinado de forma anticipada, con la finalidad de asegurar el logro de los objetivos de la empresa.” (MINTZBERG H. Y QUINN 1993)

Con el estudio de las definiciones anteriores se puede apreciar en ellos características comunes, como son: el objetivo a cumplir en el futuro, la secuencia de pasos que se necesitan y la habilidad de encaminar o dirigir los recursos para alcanzar ese fin.

1.5.2 Enfoques de las estrategias

Existen varios enfoques para la formulación de estrategias en dependencia de la forma en que se manejan los objetivos y la toma de decisiones en una organización determinada. A continuación se muestran varios de ellos.

Enfoque Ascendente: “Las iniciativas son tomadas por diversas unidades o divisiones de la organización y luego son enviadas hacia arriba para que sean completadas a nivel corporativo.”(MARTIN 2002)

Este enfoque implica que los encargados de desarrollar la estrategia contemplen valores como son: el respeto a las ideas y personas, el reconocimiento de las necesidades para garantizar el equilibrio entre los desarrolladores y por último, la transparencia para distribuir las funciones o tomar decisiones importantes. La Red Española de Desarrollo Rural afirma que este enfoque significa que “todas las fases del programa las búsquedas de soluciones y las decisiones parten desde abajo hacia arriba.”(REDR 2007)

Enfoque Descendente

“La iniciativa la toman los ejecutivos del nivel superior de la organización, quienes formulan una estrategia a nivel más bajo.”(MARTIN 2002)

En este enfoque se recorre la pirámide desde la cúspide que son los altos directivos hasta sus niveles más bajos. En el nivel superior se determinan los objetivos a seguir por cada una de las unidades y se evalúa su desempeño.

Enfoque Interactivo

“Los ejecutivos en el ámbito corporativo y los gerentes de nivel más bajo preparan una estrategia, previa consulta entre si.” (MARTIN 2002)

Como su nombre indica, representa la interrelación entre los dos enfoques anteriores, donde recoge los objetivos generales de la organización y el conocimiento de los directivos.

Enfoque a nivel dual

“La estrategia es formulada de manera independiente en el nivel corporativo y de negocios.” (MARTIN 2002)

Una primera apreciación de este enfoque puede parecer muy similar al nivel dual sin embargo, en este los planes se elaboran en función de las necesidades particulares de cada unidad dentro de la organización y los directivos se limitan a revisar dichos planes.

La selección del enfoque ha utilizar está fuertemente relacionado con el propósito que da vida a la concepción de una estrategia, sus objetivos y características propias de la organización para la cual se diseña y debe ayudar en la toma de decisiones por parte de sus directivos en cualquiera de los niveles.

1.5.3 Niveles de Estrategia

Luis Miguel Soto, Máster en Administración Estratégica de Empresas, plantea que para definir una estrategia en una organización se debe “definir en que nivel de la organización se va a aplicar la estrategia y a quienes va a implicar”(LAGUERRE 2007). El autor sigue los principios de Thompson y Strickland para definir los niveles, ellos son:

Nivel 1: Estrategia Corporativa. “Esta estrategia la formula la alta administración con el fin de supervisar los intereses y las operaciones de organizaciones que cuentan con más de una línea de negocios.”(MARTIN 2002) En este nivel la organización corporativa decide los negocios en los cuales se va a involucrar, los que puede eliminar o definir lo que realmente espera obtener de cada negocio, así como la forma en que debe distribuir los recursos para alcanzar los objetivos trazados.

Nivel 2: Estrategia de Negocio. “Es formulada para alcanzar las metas de negocios específicos y se ocupa de la administración de los intereses y operaciones de un negocio particular”. (MARTIN 2002) Este nivel se limita solo a un negocio, define cómo lo debe manejar y los productos que va a desarrollar. Según Luis Miguel Soto, “se aplica en organizaciones donde existe una línea de negocios y el éxito radicará en desarrollar una posición competitiva más poderosa a largo plazo, por ejemplo desarrollar estrategias para aumentar las ventas en un año en 10%(abarca prácticamente a todas las áreas que conforman el ciclo operativo de la empresa)” (LAGUERRE 2007)

Nivel 3: Estrategia Funcional. “Es formulada para un área funcional específica como un esfuerzo para llevar a efecto la estrategia de la unidad de negocio”. (MARTIN 2002) Principalmente se centra en determinados segmentos dentro del negocio, como por ejemplo los recursos humanos o la investigación y desarrollo. Además, este nivel está “orientado a más jefaturas, la clave está en un manejo adecuado de una actividad funcional o algún proceso importante dentro de la organización” (LAGUERRE 2007)

En el anexo 2 se puede apreciar la pirámide organizativa de los niveles de estrategia mencionados. Estos niveles no funcionan de manera independiente, sino que su relación es bidireccional y pueden coexistir varios de ellos en una misma estrategia.

1.6 Transferencia tecnológica y del conocimiento

El avance tecnológico está convirtiendo al conocimiento en el recurso más importante y dominante, de ahí que se identifique a esta generación como “Sociedad del Conocimiento”, donde la transferencia del conocimiento juega un papel decisivo en el aumento de las innovaciones en el sector empresarial.

Virginia Patricia Zozaya, de la universidad Blas Pascal de Argentina define la transferencia como “un proceso sistémico con flujos diversos y direcciones de circulación del conocimiento y con la participación de múltiples agentes. El objetivo de transmisión en la sociedad no es sólo informar y difundir, sino también transformar el entorno en la búsqueda constante del quehacer innovador y creativo que ofrece – por supuesto - la multidisciplinariedad.” (ZOZAYA 2006) La autora afirma que la transferencia de tecnología se encuentra estrechamente relacionada con la del conocimiento.

La Pontificia Universidad Javeriana de Cali, en Colombia, define la transferencia tecnológica como “un proceso para concebir un uso nuevo para una tecnología existente. También se ha presentado como un proceso para la investigación que se convierte en desarrollo económico.” (JAVERIANA 2006)

Por tanto, la transferencia tecnológica no es más que el proceso que se lleva a cabo para transferir el conocimiento necesario para la fabricación de un producto, la aplicación de procesos o prestaciones de servicios; con el fin de potenciar la capacidad, aprovechamiento y conservación de los recursos con que se cuenta, así como la comunicación y retroalimentación de los responsables de realizar la transferencia. El éxito o fracaso del proceso de transferencia depende de la propia organización, la importancia que se le brinde y el impacto en sus trabajadores.

1.6.1 Transferencia de un Modelo de Factoría de Software

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 siguiente se explica el proceso de transferencia de un modelo para replicar una Factoría de Software.

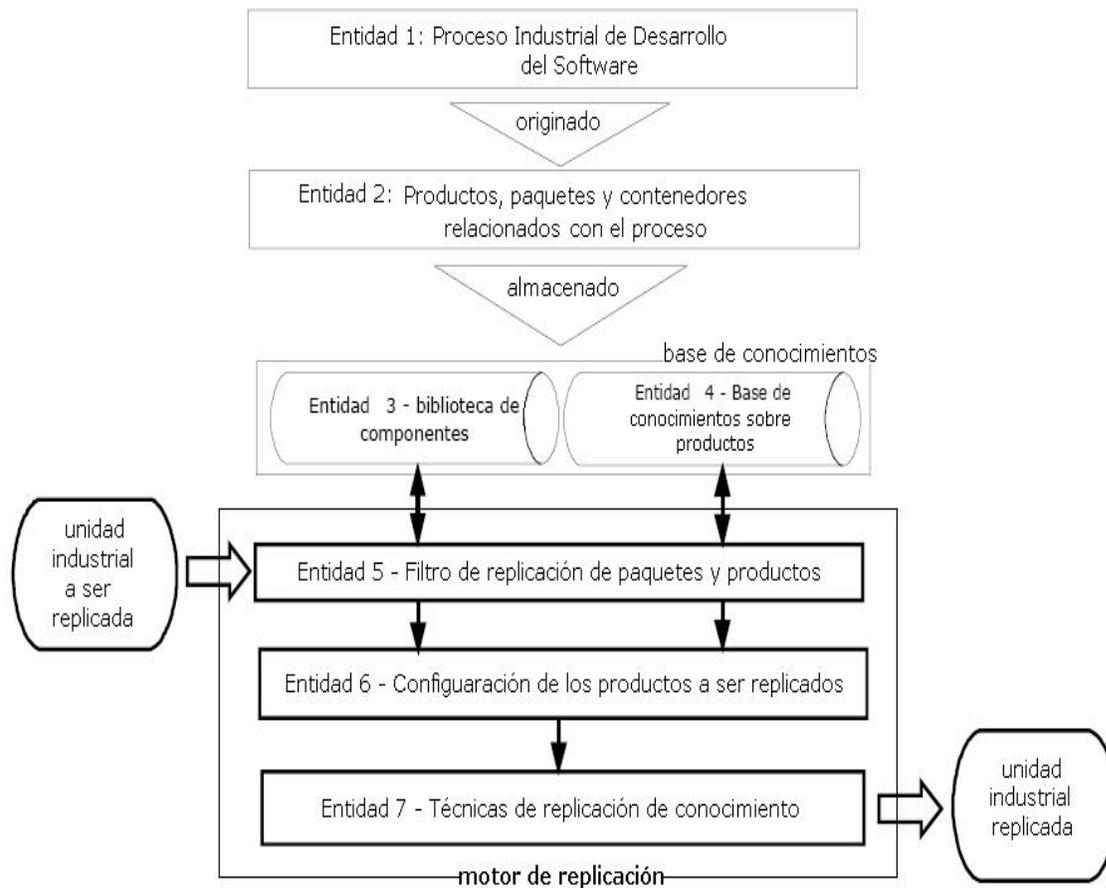


Figura 1.8 Arquitectura de Transferencia de un modelo de Factoría de Software

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.

1.6.2 Estrategias de transferencia

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.

La transferencia durante el proceso de ejecución del proyecto son aquellas que 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, etc., o productos de socialización como cartillas, entre otros”(ACEVEDO 2005)

La transferencia del producto final son aquellas estrategias que “se definen para ser implementadas una vez concluido el proceso de investigación con el propósito de difundir los resultados del mismo.” (ACEVEDO 2005)

Por tanto, 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.

1.7 Metodologías de desarrollo de Software

Las metodologías de desarrollo de software tienen la misión de guiar y controlar el proceso de producción de software, con el objetivo de garantizar la calidad y fiabilidad de los mismos. Se dividen en dos grandes grupos: las Fuertes o Establecidas y las Ligeras o Ágiles. En el anexo 3 se pueden observar varias de las características que distinguen a estos dos grupos.

Una de estas metodologías es Programación Extrema (Extreme Programming, XP). Según Patricio Letelier, especialista de la Universidad Politécnica de Valencia, “XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico.” (PATRICIO LETELIER 2004)

Las Establecidas se centran en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir y las herramientas y notaciones que se usarán. Dentro de estas metodologías se destaca RUP (Rational Unified Process). Jacobson en su libro “El Proceso Unificado de Software” utiliza para esta metodología el Lenguaje Unificado de Modelado (UML) y se basa en tres ideas básicas: guiado por casos de uso, centrado en la arquitectura y el desarrollo es iterativo e incremental.

Específicamente para el desarrollo de productos multimedia se han creado otras metodologías, entre ellas están: Relationship Management Methodology (RMM). Sasha Valdez Jiménez en su trabajo de diploma plantea “El método RMM fue la primera metodología para el diseño de multimedia. Se trata, probablemente, del único método para hipermedia que parece cubrir todo el ciclo de desarrollo, desde el estudio de factibilidad hasta la evaluación del sistema, aunque sólo propone actividades y productos concretos para las fases de análisis y de diseño.” (JIMENEZ 2005)

RMM se utiliza para el desarrollo de hipermedia que involucra grandes equipos de desarrolladores en extensos períodos de tiempo. Constituye una metodología tentadora porque define claramente las fases de la producción, además incorpora una serie de diagramas, principalmente en el diseño y la navegación.

En el desarrollo de software no existe una metodología universal sino que toda metodología debe ser adaptada al contexto del proyecto. Después de analizar las principales metodologías de la bibliografía encontrada, se propone para la línea de producción de contenidos educativos utilizar RUP. Entre los factores que influyen en su selección se destaca que cuenta con la herramienta líder en el mundo de modelación visual para los procesos de modelación del negocio, análisis de requerimientos y diseño de arquitectura de componentes, desarrollada por la corporación Rational y que cubre todo el ciclo de vida de un producto de software.

1.8 Estándares de Calidad

A lo largo de la historia, el hombre en su afán de perfección ha visto la necesidad de asumir responsabilidades que revisen de forma sistemática la efectividad de su trabajo. En el mundo de la informática, la efectividad se mide en términos de calidad del proceso de desarrollo y de los productos en general. Según Pressman, “la calidad del software se define como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.” (PRESSMAN 2002)

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto software o la calidad de su proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos a establecer para la calidad del proceso de desarrollo.

Un buen proceso de desarrollo está dirigido por estándares de calidad. Según la Fundación SES, “conocer los estándares organizacionales le permite a una organización poder pensarse: revisar su estructura, su funcionamiento, sus relaciones, entre otros, saber con mayor profundidad, cuáles son sus fortalezas y cuáles sus debilidades, de manera de poder planificar estrategias que posibiliten su fortalecimiento y desarrollo.” (EDCO 2002)

En este epígrafe se presenta una selección de los modelos y estándares de calidad más representativos encontrados en la bibliografía consultada y que han sido reconocidos y adoptados por diferentes empresas y entidades, principalmente aquellos que son aplicables al enfoque de Factoría de Software.

1.8.1 Capability Maturity Model Integration (CMMI)

El modelo CMMI se originó a partir de varios modelos, entre ellos se encuentra CMM (Capability Maturity Model) publicado en 1991 por el Software Engineering Institute (SEI) y que estaba orientado a la mejora de los procesos relacionados con el desarrollo de software, para lo cual contempla las consideradas mejoras prácticas de ingeniería y gestión de software.

El Grupo CIENTEC fue fundado en 1977, siendo líder en el mercado de sistemas de computación personal afirma que “CMMI integra un conjunto de mejores prácticas basadas en varias disciplinas, que abarcan el diseño y análisis de sistemas, ingeniería de software, desarrollo integrado de productos y gestión de proveedores para proyectos. Con él, una organización puede implementar simultáneamente una serie de mejoras que, de otra manera, serían consideradas como iniciativas separadas.” (CIENTEC 2007)

Además de integrar varias disciplinas, clasifica las empresas en niveles de madurez. CMMI, define una escala jerárquica desde el nivel 0 hasta el 5, que representan el incremento en las capacidades de las áreas de proceso en mejoramiento. Julio Ariel Hurtado, especialista de la Universidad de Cauca explica que “el escalón más bajo de la escala denota que la ejecución del proceso no cumple con el propósito del mismo, poniendo en riesgo la calidad del producto por él generado, mientras que el nivel más alto indica que su ejecución cumple ampliamente los objetivos del negocio, asegurando así la satisfacción y calidad del producto de software generado.” (JULIO ARIEL HURTADO 2005)

Este estándar permite a la empresa organizar su proceso de forma que sea efectivo, controlado y mejorado con varias de las disciplinas de ingeniería y gestión de software entre otras, por lo que su aplicación dentro de una Factoría de Software implicará en la mejora continua de los indicadores de calidad, productividad, planificación, costo y satisfacción de los clientes con el cumplimiento de los requisitos planteados.

1.8.2 International Standard Organization (ISO)

Los principios básicos de la gestión de la calidad están encaminados a mejorar el funcionamiento de una organización. Para esto se han creado normas, que combinándose con los principios técnicos se logra una mayor satisfacción del consumidor. Una de estas normas es la ISO 9001: 2000 está dirigida a los sistemas de gestión de la calidad (SGC).

La Secretaría de Energía de Buenos Aires en su informe del 2003 define que esta norma “especifica los requisitos para los SGC aplicables a toda organización que necesite demostrar su capacidad para

proporcionar productos que cumplan los requisitos de los clientes y los reglamentarios que le sean de aplicación y su objetivo es aumentar la satisfacción del cliente.”(PIEEP 2003)

El Dr. Mark C. Paulk afirma en 1993 en el diario de calidad del software que, “la ISO 9001 requiere que la política de calidad esté definida, documentada, sea comprendida, implementada, y mantenida. Un gerente designado asegura que el programa de calidad sea implementado y mantenido.” (M.C. PAULK 1993)

Según estos autores se puede asumir que cuando una organización adopta el enfoque de esta norma genera confianza en la capacidad de sus procesos, en la calidad de sus productos y proporciona las bases para la mejora continua, debido a que exige requisitos de organización y define las actividades que le permitan verificar sistemáticamente el cumplimiento de la calidad prevista y la satisfacción del cliente. Esta norma propone hacer mediciones, análisis y mejoras mediante supervisiones y auditorías internas en la organización, que inspeccionen el servicio y brinden una mejora continua del producto y el proceso.

El modelo de sistema de gestión de la norma IRAM-ISO 9001:2000 se puede resumir en los siguientes pasos:

- Planificar: Establecer los objetivos y procesos necesarios para conseguir resultados de acuerdo con los requisitos del cliente y las políticas de la organización.
- Hacer: Implementar los procesos.
- Verificar: Realizar el seguimiento y la medición de los procesos y los productos respecto a las políticas, los objetivos y los requisitos para el producto e informar sobre los resultados.
- Actuar: Tomar acciones para mejorar continuamente el desempeño de los procesos.

En el anexo 4 se puede observar el proceso que propone la norma para la mejora continua en el cumplimiento de los requisitos del cliente. Es aplicable tanto a productos como a servicios.

1.8.3 PSP (Personal Software Process) y TSP (Team Software Process)

Expertos de la Universidad Aeronáutica de Embry-Riddle en el prólogo al libro “Introducción al Proceso Personal de Software”, defienden que “para desarrollar productos software con calidad y de una forma eficiente y eficaz, la industria depende en gran parte de la capacidad individual de los ingenieros.” (HUMPHREY, WATTS S. 2001a)

Por esto es que surge el Proceso de Software Personal (PSP), definido por Watts S. Humphrey del Software Engineering Institute (SEI) en la Carnegie Mellon University. “El Proceso Software Personal (PSP) fue diseñado para ayudar a los ingenieros del software a hacer bien su trabajo. Muestra como

aplicar métodos avanzados de ingeniería a sus tareas diarias. Proporciona métodos detallados de planificación y estimación, muestra a los ingenieros como controlar su rendimiento frente a estos planes y explica cómo los procesos definidos guían su trabajo.” (HUMPHREY, WATTS S. 2001a)

El Proceso de Software en Equipo (TSP), tiene la misión de guiar el trabajo en equipos basado en los principios del PSP. Humphrey plantea al respecto: “El proceso del equipo de software (TSP, Team Software Process) es un proceso que al igual que el PSP, está basado en el modelo CMM. TSP está diseñado para ayudar a controlar, administrar y mejorar la forma en que trabaja un equipo de software. Al igual que PSP, está estructurado por formularios, guías y procedimientos para desarrollar software. “(HUMPHREY, WATTS S 2001b)

El TSP es un método para la mejora del trabajo en equipos que se dedican a la producción de software. Cada desarrollador tiene definido sus roles, sus actividades y sus responsabilidades. Además incluye procedimientos para la mejora continua del proceso de desarrollo, para mejorar la calidad del software producido, la estimación del tiempo de desarrollo, la disminución de defectos en el producto y para promover la integración del equipo de desarrollo. Apoya tanto al equipo de desarrollo como a los administradores del proyecto para la culminación a tiempo y dentro del presupuesto de los proyectos de desarrollo de software.

La utilización de TSP en un proyecto resuelve problemas como: la falta de comunicación, de organización y desconocimiento de los roles. A su vez, PSP ayuda al ingeniero a ser más eficiente, planificando su trabajo personal de acuerdo a estimaciones realizadas con sus propios datos históricos y esforzándose por obtener productos con mejor calidad. Muestra a los implicados en el proceso cómo controlar su rendimiento basado en un proceso de mejora continua.

1.9 Tecnología multimedia

A pesar de que con el surgimiento de las TICs se ha popularizado el término Multimedia, el mismo no es nuevo. José Luis Ulizarna García, especialista de la Universidad de Valencia explica su punto de vista, “la multimedia se encuentra en el punto medio entre los medios editoriales tradicionales (texto, gráficos, fotografías) y el medio audiovisual (animaciones, sonido y vídeo) dado que emplea ambos de forma entrelazada. Es pues un medio capaz de integrar texto, imágenes (estáticas o dinámicas), sonidos y voz dentro de un entorno único”. (GARCÍA 1998)

Según Rodríguez del Instituto Tecnológico de Celaya, una multimedia “es una nueva plataforma donde se integran componentes para hacer ciertas tareas que proporcionan a los usuarios nuevas oportunidades de trabajo y acceso a nuevas tecnologías. Es un nuevo medio donde la computadora junto con los medios tradicionales da una nueva forma de expresión. Es una nueva experiencia donde la interacción con los medios es radicalmente diferente y donde se hace necesario aprender como usarlos. Es una nueva industria donde, con una nueva plataforma, un nuevo medio y una nueva experiencia, nos llevan a tener nuevas oportunidades de negocios.” (RODRÍGUEZ 2005)

El término multimedia, de acuerdo con los criterios anteriores, se refiere a la unión de varias medias, principalmente texto, imagen, animaciones, audio y video. Tiene varias aplicaciones, por ejemplo en los juegos para diversión y entretenimiento, en los negocios, en la difusión del saber y del conocimiento por solo mencionar algunas.

1.10 Tecnologías y plataformas de desarrollo actuales

Las herramientas que se utilizan para el desarrollo de software determinan su éxito en el mercado actual. Una correcta selección de las tecnologías es lo que permite el buen funcionamiento de los productos. Es por ello que en este epígrafe se presentarán algunas de estas tecnologías con sus características distintivas, para justificar así las que se utilizarán en la línea de producción de contenidos educativos para el modelo de factoría propuesto.

1.10.1 Tecnologías para sitios Web

HyperText Markup Language (HTML)

El lenguaje de publicación usado por la World Wide Web (www) para publicar información que todos los dispositivos puedan comprender es el HTML (HyperText Markup Language). Tiene un carácter estático y no interactúa con el usuario. Necesita un navegador para que se pueda interpretar.

La versión 4 de este lenguaje desarrolla mecanismos para hojas de estilo, ejecución de scripts, marcos y objetos incluidos. Permite publicar documentos, textos, fotos, tablas, soporta la navegación a través de hipertextos y presenta mejoras en el trabajo con formularios. Nicolás Escobar Jariton de la consultoría informática de México, afirma que “conocer en plenitud este "lenguaje de marcado", permite utilizarlo en combinación con otros lenguajes de programación.” (JARITON 2007)

Es importante conocer con profundidad este lenguaje a pesar de que ya existan otros, debido a que representa un estándar para la creación de páginas Web, está en formato texto y sirve para colocar los objetos en la página con las características indicadas.

Extensible Hyper Text Markup Language (XHTML)

El lenguaje de marcado XHTML se creó con el objetivo de sustituir el HTML como estándar para las páginas Web. Nicolás Escobar Jariton afirma que “el XHTML es una reformulación del HTML4 para cumplir los estándares y así hacerlo más portable, sin tener que crear otro lenguaje totalmente diferente para hacerlo.” (JARITON 2007) XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo y JavaScript su aspecto y diseño en distintos medios.

Algunos navegadores antiguos no son totalmente compatibles con los estándares, lo que hace que las páginas no siempre se muestren correctamente y muchas herramientas de diseño Web aún no producen código XHTML correcto.

Extensible Markup Language (XML)

Maria Isabel García Arenas especialista de la Universidad de Granada define que “XML no es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.” (ARENAS 2007)

XML presenta todas las ventajas del HTML añadiendo la capacidad de describir de forma estructurada el contenido del documento de manera independiente a la definición del formato de presentación de dicha información. Es una arquitectura abierta y extensible para cualquier navegador. La gestión de los datos se puede hacer desde el propio cliente Web, los motores de búsqueda pueden devolver datos más precisos y la información resulta más accesible.

Cascading Style Sheets (CSS)

Nicolás Escobar Jariton plantea que “las hojas de estilo facilitan muchísimo la creación de páginas. Gracias a éstas se puede definir un estilo específico para cada atributo sin tener que modificar las

etiquetas dentro de la página HTML. Por ejemplo, se puede utilizar un solo archivo para manejar los estilos de todo un sitio.” (JARITON 2007)

Las hojas de estilos en cascada (CSS) constituyen un lenguaje sencillo que complementa el del HTML para transformar las etiquetas del lenguaje HTML a las características que el diseñador desee. Son un conjunto de reglas de formato que controlan el aspecto del contenido de una página Web. Los estilos CSS aportan flexibilidad y control al aspecto exacto que se busca en una página, en cuanto a posición, el diseño de fuentes y estilos concretos. Se utilizan para combinar una serie de atributos del texto, como pueden ser el color o el tamaño de todos los documentos de un sitio Web desde un archivo.

JavaScript

Es un lenguaje basado en objetos y orientado a eventos. Es débilmente tipificado ya que no es necesario declarar el tipo de datos y se realizan conversiones entre ellos automáticamente cuando es necesario y si es viable. Víctor Rivas Santos, especialista de la Universidad de Jaén, afirma que “JavaScript se ejecuta tanto en los navegadores de Netscape como en los Internet Explorer de Microsoft (NS e IE respectivamente).” (SANTOS 2007)

La característica principal de JavaScript es la de ser un lenguaje interpretado como idóneo para trabajar en Web, ya que son los navegadores del cliente los que interpretan y ejecutan los programas escritos en JavaScript que se integran a páginas HTML. Los documentos que contengan código Javascript dejan de ser estáticas para convertirse en dinámicos.

1.10.2 Plataformas de desarrollo Web

FrontPage

Frontpage es una herramienta creada por Microsoft para editar sitios Web, es ideal para personas inexpertas y sin conocimientos de HTML. Sus capacidades son semejantes a las de otros editores. Es una aplicación cliente/servidor. Entre sus funcionalidades está la de facilitar la edición y diseño de páginas html. Dentro de las posibilidades que brinda se observa el reconocimiento de extensiones html para Internet Explorer y permite trabajar con marcos y tablas. Además, Senso en su artículo para la revista Internacional Científica y Profesional, afirma que “tiene la opción de Wysiwyg (what you see is what you get) para la comprobación de los documentos html.” (SENSO 1996)

Una desventaja de este paquete es que no es un software propietario, además de que está diseñado para el navegador de Microsoft y otros navegadores pueden presentar problemas al cargar los sitios realizados.

Dreamweaver

Dreamweaver es considerada la herramienta de diseño de páginas Web más avanzada, puede utilizarse para diseñar páginas profesionales y proporciona una potente combinación de herramientas visuales de diseño, funciones y soporte para la edición del código, además de las necesarias para un entorno integrado y agilizado. Soporta gran cantidad de tecnologías fáciles de usar, como hojas de estilo y capas. Permite la inserción de archivos multimedia y crear efectos e interactividades. A partir de su versión 4, ya incluye soporte para la creación de páginas dinámicas de servidor, con acceso a bases de datos y una mayor integración con otras herramientas de Macromedia como Fireworks.

Jen Taylor, jefa de producto para Dreamweaver en la compañía Macromedia plantea que desde su lanzamiento en 1997, “Dreamweaver de Macromedia se ha convertido en la solución estándar de la industria para el desarrollo profesional de la Web y actualmente, más de 3,2 millones de profesionales utilizan Dreamweaver para desarrollar sus sitios Web.” (TAYLOR 2005)

Como el objetivo del proyecto es desarrollar productos multimedia orientados a la Web, se hace necesario tener en cuenta la utilización de este editor de páginas Web.

Flash

Flash es una de las herramientas más populares para crear presentaciones multimedia para la Web, por la utilización de imágenes vectoriales, reutilización de recursos y animación interpolada para minimizar espacio. Además de permitir gráficos y animaciones. Flash posibilita la inserción de sonidos e importar otros formatos, así como agregar programación e interacción.

Brenda M. Michelson, Vicepresidenta primera y consultora principal de Patricia Seybold Group plantea que “más de 1 millón de diseñadores y desarrolladores están trabajando con la plataforma Flash hoy en día.” (MICHELSON 2005)

Esta plataforma es compatible con muchas otras plataformas y su arquitectura permite ofrecer contenido, aplicaciones dinámicas y comunicaciones en una variedad de puntos de contacto (navegador, dispositivo

móvil, cliente de escritorio, receptores digitales multimedia, etc.) que representan una amplia gama de sistemas operativos, factores de forma y proveedores.

1.10.3 Selección de las herramientas a utilizar

La amplia variedad de tecnologías existentes para el desarrollo de productos para la Web constituye un factor importante para decidir qué herramienta utilizar.

Teniendo en cuenta que los productos que la factoría debe elaborar son multimedia para la Web, se propone trabajar con Dreamwaver por las facilidades que brinda, básicamente utilizando HTML, CSS y Javascript para la integración de los elementos de multimedia y la creación de los contenidos educativos, es perfectamente compatible con la plataforma Flash que se utilizará solamente para las actividades interactivas de los productos. No se recomienda utilizar FrontPage debido a que no es confiable para visualizar sus productos con navegadores diferentes al Internet Explorer, característica sumamente importante si se tiene en cuenta que los productos a desarrollar son para el país de Venezuela, que se encuentra en fase de emigración total a software libre y plataforma Linux.

1.11 Conclusiones

En este capítulo se han definido los conceptos fundamentales para el desarrollo de la investigación científica en la obtención de la estrategia de transferencia de un modelo de Factoría de Software para la línea de Contenidos Educativos, concluyendo en este sentido que el enfoque de Factoría de Software busca perfeccionar el proceso actual aplicando principios de ingeniería de software a la producción.

Se analizaron varios modelos donde cada uno presentaba características importantes pero como principal desventaja que no tenían en cuenta la gestión de proyectos, sino que están más bien orientados a perfeccionar la producción. Se escogió el Modelo de Factoría de Software aplicando Inteligencia por considerarse el más completo de todos e incluir los aspectos más significativos de los otros modelos.

Este modelo recoge la experiencia del modelo basado en la norma ISO 9001 con el uso de estándares internacionales como ISO, propone que se utilice CMMI en sustitución de CMM debido a que este último ya fue abolido e incluye PSP y TSP en la planificación del trabajo a nivel personal y de equipo para elevar la calidad del proceso de desarrollo y del producto final, mediante un proceso repetible y mejorable continuamente. Otra característica que tienen en común es que definen las bases tecnológicas para la gestión, soporte y construcción del producto. También organiza las personas por roles, permitiendo la

especialización de las funciones y la consolidación de los conocimientos basados en la experiencia de sus trabajadores, como en el modelo propuesto por Basili. Otro aspecto importante es que clasifica la factoría según el alcance dentro de la metodología utilizada y enfatiza en la gestión de proyectos, aspecto importante que la mayoría de los modelos dejan sin definir. Ninguno de los modelos incluye el tratamiento de la orientación estratégica y este modelo lo explica en la entidad Centro de Inteligencia, aunque no es de interés implantar esta entidad en un primer momento de la factoría porque se priorizarán los elementos que más afectan la producción.

Además, se mencionaron las experiencias que han tenido algunas empresas trabajando con el enfoque de Factoría de Software, principalmente en la reducción de los costos de producción, tiempo de desarrollo y la satisfacción de sus clientes por el aumento de la calidad del producto.

Se estudiaron conceptos relacionados con la estrategia de transferencia, secuencia de pasos lógicos para alcanzar los objetivos propuestos. Se escogió el enfoque interactivo para facilitar la comunicación durante el proceso de transferencia dentro del nivel funcional que es donde se enmarca el proyecto como unidad de negocio. Llevar a la práctica un modelo de Factoría de Software implica también un proceso de transferencia tecnológica y de conocimientos que debe ser apoyada por todos los trabajadores de la organización que lo van a implementar, por lo que la estrategia de transferencia del modelo seleccionado se debe aplicar durante el proceso de ejecución del proyecto.

Se investigó sobre varias metodologías de desarrollo, principalmente entre la XP y RUP, inclinándose la selección hacia este último. También en este capítulo se analizaron varios estándares de calidad que permiten obtener un buen proceso de desarrollo.

Finalmente se escogieron las herramientas y tecnologías que se utilizarán en el desarrollo de las multimedia de tipo Web, considerando que el modelo de factoría se define para la línea de producción de contenidos educativos.

CAPÍTULO 2: ESTRATEGIA DE LA TRANSFERENCIA

2.1 Introducción

En este capítulo se describe el proceso de desarrollo de software educativo en la Universidad de las Ciencias Informáticas (UCI), los métodos, procedimientos y técnicas utilizadas en la investigación. Se define la estrategia de transferencia del modelo de Factoría de Software aplicando Inteligencia y los riesgos que puedan afectar el éxito de la misma.

2.2 Proceso de desarrollo de software educativo

El desarrollo de la industria cubana del software se encuentra estrechamente relacionado a la UCI, la misma constituye un nuevo modelo de formación – investigación – producción en el campo de las TIC y su impacto se hace sentir ya en diferentes sectores de la sociedad y la economía nacional.

La universidad cuenta con una Infraestructura Productiva (IP) para organizar la producción y dirigir metodológicamente los proyectos. Cuenta con una serie de direcciones para las líneas de producción, gestión y calidad. Las líneas de producción se nutren según los perfiles de cada una de las facultades, algunos ejemplos son: Informática Educativa y Multimedia, Bioinformática y Realidad Virtual.

Específicamente en la línea de producción de Informática Educativa y Multimedia, los proyectos desarrollan productos para diferentes entidades en el marco nacional y en el exterior del país, como es el caso de CNTI – Contenidos Educativos.

La estrategia adoptada en la producción de los productos de software educativo y multimedia en la UCI para garantizar la adecuada ejecución de los proyectos consta de un flujo general de trabajo que se divide en seis procesos (YADENIS PIÑERO 2006), una imagen gráfica del flujo se puede encontrar en el anexo 5:

- Definición del proyecto.
- Gestión de requisitos y análisis.
- Evaluación técnica.
- Gestión de medias.
- Gestión de diseño gráfico.
- Construcción.

El primer proceso es la definición del proyecto, etapa de concepción inicial del producto, donde el cliente entrega su solicitud y el guión de contenido (si lo tiene elaborado) y se define la estrategia de trabajo.

El proceso de Gestión de requisitos y análisis resulta de vital importancia ya que aquí se realiza una descripción detallada del objeto de estudio, se analiza la solicitud del cliente y del guión de contenido y se aclaran las posibles dudas sobre la documentación entregada. Se revisa el guión hasta tener una total comprensión del mismo y cuando se logra esto se elabora el guión técnico. Debe quedar clara la necesidad de elaborar el producto, el público al que va dirigido, los objetivos pedagógicos que se pretenden cumplir, los contenidos a tratar y los medios para presentarlos.

En el proceso Evaluación Técnica se realiza un análisis del proyecto en cuanto a las herramientas que se utilizarán para el desarrollo, los equipos de trabajo que se conformarán, el hardware necesario tanto para realizadores como para usuarios, la factibilidad técnica y económica de su producción (presupuesto necesario), las formas de distribución, las especificaciones de los productos a elaborar y se realiza una recomendación de la arquitectura a usar para la producción de software educativo así como de la arquitectura organizativa para acometer la producción.

La gestión de medias comprende los procesos de búsqueda o producción de determinados recursos audiovisuales en función de los requisitos dados en el guión técnico. Esta tarea es responsabilidad del equipo de audiovisuales de la Dirección de Comunicación Audiovisual.

La gestión de diseño gráfico comprende los procesos de concepción y realización del diseño gráfico de un producto de software educativo en función de los requisitos dados en el guión técnico. En esta etapa se obtendrá una información detallada de cómo estará estructurado el programa y el diseño de la interfaz de cada una de las pantallas por parte del equipo de diseñadores de la Dirección de Diseño Visual.

En el proceso de Construcción se cumplen dos tareas de singular importancia: la obtención y edición de todos los medios que serán empleados y la programación por parte de los proyectos en cada facultad. La implementación se rige por el guión técnico, donde se realizan revisiones técnicas para corregir posibles errores antes de ser entregado al cliente.

Sin embargo, muchos de estos proyectos tienen deficiencias en todas las etapas de desarrollo lo que afecta la calidad de los productos. Uno de los proyectos es CNTI – Contenidos Educativos, quien está a cargo de elaborar gran cantidad de productos multimedia para la Web. Este proyecto surgió como parte del apoyo que brinda la Universidad de las Ciencias Informáticas (UCI) al Convenio Integral firmado entre Cuba y Venezuela, donde participan el Centro Nacional de Tecnologías de Información de Venezuela (CNTI) y la empresa cubana Sistemas Informáticos de Software SIS – Copextel.

Su trabajo depende mayormente de las entregas del equipo de diseño gráfico de la IP. El diseño gráfico en ocasiones no se parece al guión entregado y como no se documenta, el equipo de desarrollo tiende a perderse al integrar los contenidos. Se trabaja desorientadamente y el producto se entrega con defectos, que para corregirlos se atrasan los plazos convenidos con el cliente. A esto también se le suma, la inexperiencia de muchos de sus líderes y el desconocimiento de las herramientas para el desarrollo por parte de los integrantes del proyecto. Se realizan malas planificaciones del trabajo, las estimaciones del tiempo no se basan en datos reales, provocando incumplimientos en el tiempo de entrega y los componentes desarrollados no se documentan para facilitar su reutilización en futuros proyectos.

Todos estos problemas van empeorando a medida que aumenta la fuerza de trabajo y la creciente demanda de producción de software educativo, implica desorganización de la producción y disminución de la productividad. En ocasiones se corre el riesgo de que los proyectos fracasen o que el cliente no se sienta satisfecho con el producto entregado.

Una solución a estos problemas puede ser la implantación del enfoque de Factoría de Software al proyecto CNTI – Contenidos Educativos, ya que el objetivo de este enfoque es industrializar el desarrollo de software, definiendo los procesos de desarrollo con métodos y técnicas estandarizadas para la mejora continua; agrupadas por líneas de producción, donde cada trabajador asume una tarea específica y centra sus esfuerzos en ella. En una Factoría de Software el tiempo de desarrollo y el costo de producción disminuyen con la reutilización de componentes, brindando también una garantía de la calidad del producto final. Se logra un buen entendimiento de lo que se necesita hacer a través de la interacción con el cliente, buscando con ello que el cliente se sienta satisfecho con la entrega final. Las estimaciones del tiempo son precisas, ya que están basadas en el conocimiento real de la capacidad productiva según datos históricos. Además se brinda capacitación a los recursos humanos implicados en el proceso.

2.3 Métodos, procedimientos y técnicas utilizados

Todo método científico de investigación necesita una estrategia para abordar las características y condiciones específicas del problema a resolver. La estrategia para esta investigación fue descriptiva y se desarrolló en relación al proyecto CNTI – Contenidos Educativos, dirigida fundamentalmente a las personas involucradas en los procesos de desarrollo de contenidos educativos.

Se utilizaron métodos teóricos para estudiar las características del objeto que no están observables directamente, elaborar una hipótesis y proponer mejoras. Estos métodos son: Histórico lógico, Hipotético deductivo y Sistémico.

Se realizó una profunda revisión bibliográfica para recopilar los aspectos necesarios para determinar la necesidad de implantar la Factoría de Software.

El método histórico lógico permitió investigar todo lo relacionado con este enfoque, en cuanto a conceptos, objetivos, características, funciones, como se está aplicando en otras empresas y que resultados han experimentado.

El hipotético deductivo se encuentra presente en el cumplimiento del objetivo y la hipótesis planteada.

Un papel fundamental en la investigación lo tienen los métodos empíricos porque brindan la información necesaria sobre el objeto de estudio. Las encuestas y entrevistas realizadas al personal involucrado en el proceso de elaboración de los contenidos educativos en el proyecto, permitieron conocer su funcionamiento, organización y soluciones.

Se decidió aplicar la técnica de muestreo no probabilística a una población determinada, ya que esta técnica permite ahorrar tiempo y recursos. Se eligió un muestro no probabilística de juicio o criterio, mediante la aplicación de encuestas, por lo que se seleccionaron los miembros de la población que serían encuestados, para fundamentar la necesidad de implantar la Factoría de Software. Se aplicaron encuestas a los integrantes del proyecto productivo CNTI – Contenidos Educativos, de la facultad 8, con el objetivo de valorar la situación actual por la que transitaba el proyecto. Se midieron aspectos relacionados con la organización y formas de trabajo, elementos fundamentales de una Factoría de Software. De un total de 35 integrantes del proyecto se aplicó encuestas a 25 desarrolladores, siendo la muestra un 72 % de toda la población.

En las encuestas aplicadas se evaluaron indicadores relacionados con el flujo de procesos del desarrollo de los productos multimedia, como son: metodologías de desarrollo, planificación del trabajo, tecnologías y herramientas de trabajo, mecanismos de calidad y estructura organizacional del proyecto.

Para elaborar las encuestas aplicadas se tuvo en cuenta las variables o indicadores que se querían medir y el objetivo que se perseguía con las mismas. Se hicieron preguntas cerradas y semi-cerradas mayormente, para tener un poco más de profundidad en los criterios de los encuestados. (Ver anexo 6)

Para comprobar los resultados esperados con la prueba de la estrategia de transferencia del modelo de factoría se decidió aplicar nuevamente encuestas a los miembros de la factoría. (Ver anexo 7)

Además, se realizaron entrevistas informales mediante conversaciones al líder del proyecto con el objetivo de conocer solamente un tema determinado por los investigadores. Estos intercambios proporcionaron los datos relativos a sus conductas, deseos, actitudes y expectativas, a través de un listado de preguntas que ayudaron a obtener una información relevante sobre el proyecto.

2.4 Estrategia de transferencia del modelo

Definir una estrategia de transferencia de un modelo de Factoría de Software implica poner al proyecto en posición para realizar su misión con eficacia y eficientemente. El proceso de transferencia cubre una gama de actividades directivas, incluyendo la motivación por parte de sus integrantes y una secuencia de procesos de evaluación que controlen el flujo de trabajo de la empresa.

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.

El primer paso se realiza para conocer las características del proyecto, y buscar alternativas para solucionar las deficiencias que presente. Se analizan los distintos tipos de modelos que existen y la experiencia de las empresas que respaldan sus beneficios.

El próximo paso es definir el proceso de transferencia, el mismo es orientado a procesos, y éstos son definidos atendiendo a las características y objetivos principales de las Factorías de Software. Estos procesos se configuran en paquetes, lo que facilita la reutilización para aquellas empresas que deseen aplicar los principios de Factoría de Software.

La entidad Gestión de Proyectos del modelo de factoría aplicando Inteligencia es la responsable de definir el proceso a partir de una metodología. El proceso depende de la clasificación de las factorías según los flujos de trabajo por los que transite (Ver Anexo 8).

El proceso de desarrollo que se realiza en el proyecto comienza a partir de la entrega del diseño para la construcción, por lo que para la línea de Contenidos Educativos la factoría se clasifica como de implementación de software, esta es la menor de las clasificaciones porque solo se dedica a implementar y probar.

Por lo tanto según lo visto por José Augusto Fabri en el capítulo 1, se deben definir todos los procesos que existen en la factoría y agruparlos en paquetes teniendo en cuenta criterios de funcionalidad, así se replicarán paquetes que contengan procesos con funcionalidades similares. Los paquetes son definidos de acuerdo al orden de implantación y cada uno se corresponde con las entidades del modelo de Factoría de Software aplicando Inteligencia. Las entidades son: Gestión de Proyecto, Proceso de desarrollo, Centro de Inteligencia, Repositorio, Personas y Bases Tecnológicas.

Tabla 2.1 Instrumentación de la Estrategia

Paquete	Procesos	Responsables	Participante
Gestión de Proyecto	Definir el proyecto y establecer la estructura organizativa de los equipos. Definir claramente los objetivos y establecer una planificación del proyecto. Realizar estimaciones concretas y reales de tiempo, costos y recursos. Controlar y supervisar los trabajos, inversiones, consumo de recursos, costo y tiempo de ejecución del proyecto. Establecer criterios de calidad sobre los resultados esperados y comprobar su cumplimiento. Prever los posibles riesgos que puedan surgir. Coordinar y supervisar las distintas tareas, cantidad de personas y actividades que se realizan en el proyecto.	- Gerente de la factoría	- Económico - Planificador - Jefe de la línea de desarrollo
Proceso de desarrollo	Identificar flujos de trabajo por fases. Determinar roles, actividades y artefactos.	- Gerente de la factoría - Gerente de desarrollo - Jefe de prueba	- Arquitecto - Programadores - Administrador del Repositorio - Revisores - Probadores
Centro de Inteligencia	Definir roles para la gestión empresarial y del conocimiento, la vigilancia tecnológica, la inteligencia empresarial y la prospectiva.	- Asesores de la factoría	- Gestor de información interna y externa - Gestor empresarial

	Gestionar el proceso de capacitación.		
Repositorio	Configurar el repositorio de componentes. Definir la estructura de almacenamiento de la información Elaborar un catálogo de componentes reutilizables. Documentar componentes: descripción, objetivo, características tecnológicas y ubicación. Definir estándares del repositorio. Actualizar los componentes existentes según los requerimientos.	- Administrador del Repositorio	- Documentadores - Gerente de la factoría
Personas	Definir roles en función del proceso. Controlar la planificación personal después de capacitarlos. Instrumentos PSP.	- Gerente de la factoría	- Planificador - Gerente de desarrollo
Bases Tecnológicas	Identificar las herramientas. Instalar en las máquinas según las necesidades de los roles. Guardar los instaladores en el repositorio. Verificar el funcionamiento en la factoría. Definir mecanismos de seguridad de la información. Establecer técnicas de configuración de herramientas.	- Gerente de soporte	- Gerente de la factoría

El proceso industrial de desarrollo de software para el mecanismo de implantación de la Factoría de Software explicado en el capítulo 1 representa el modelo de factoría aplicando inteligencia que se desea transferir al proyecto productivo. El modelo seleccionado para ser replicado se organiza en paquetes que contienen los procesos de la factoría, agrupados de acuerdo a las funcionalidades de los mismos. No se consideró necesario definir contenedores ya que los procesos son simples y se pueden agrupar solamente en paquetes. Estos paquetes se almacenan en una biblioteca de componentes donde se recoge toda la información referida a la Factoría de Software que se va a implantar.

La base de conocimientos sobre productos propone que se designe una persona con plenos conocimientos sobre todos los procesos que se desarrollan en la factoría para que asesore la transferencia de la misma.

El motor de replicación filtra todos los paquetes que serán replicados, o sea, se hace un análisis de los paquetes definidos para ver si los procesos que en ellos se agrupan cumplen con las funcionalidades previstas para la factoría. Una vez definidos y analizados los paquetes que serán replicados se configuran de forma tal que transferencia se haga más simple.

2.4.1 Gestión de Proyecto

La gestión de proyectos es la disciplina encargada de organizar y administrar recursos de manera tal que se pueda terminar un proyecto determinado en el tiempo, costo y alcance planificados. O sea, es la encargada de la planificación y control del proyecto.

Los objetivos principales de la gestión de proyectos son:

1. Definir el proyecto y establecer la estructura organizativa de los equipos.
2. Definir claramente los objetivos y establecer una planificación del proyecto.
3. Realizar estimaciones concretas y reales de tiempo, costos y recursos.
4. Controlar y supervisar los trabajos, inversiones, consumo de recursos, costo y tiempo de ejecución del proyecto.
5. Establecer criterios de calidad sobre los resultados esperados y comprobar su cumplimiento.
6. Prever los posibles riesgos que puedan surgir.
7. Coordinar y supervisar las distintas tareas y actividades de las que consta el proyecto.

El encargado de realizar todas las actividades comprendidas dentro de la gestión de proyectos es el gerente de la factoría, utilizando las técnicas y herramientas que se explicarán a continuación.

2.4.1.1 Procesos de gestión

El modelo aplicando inteligencia propone para la entidad Gestión de proyecto una serie de procesos que describen, organizan y efectúan el trabajo del proyecto, se agrupan en 6 categorías y cada una a su vez contiene una serie de actividades distintas a las de la otra categoría. Estos procesos se relacionan por sus resultados ya que los artefactos que una genera sirven como entrada para el siguiente proceso, logrando una relación iterativa y dinámica. Están fuertemente relacionados con el flujo de procesos propuesto en el paquete proceso ya que la gestión del proyecto depende completamente del proceso de desarrollo de los productos. A continuación se muestran los procesos por los que debe transitar una correcta gestión de proyecto y en el anexo 9 se explica las relaciones entre ellos:

- Proceso de Iniciación: Este proceso comienza cuando el jefe de la línea de producción se reúne con el gerente de la factoría y le asigna la responsabilidad de desarrollar los productos. Luego, se coordina una reunión con el cliente, para especificar los requerimientos que solicita.
- Proceso de Planificación: Define los objetivos del proyecto y le brinda al cliente las posibles mejoras a su solicitud. Analiza los requerimientos, las alternativas posibles y determina el alcance,

costo, riesgos y tiempo requerido para finalizar el trabajo. Se le informa al cliente la planificación estimada y si ambos están de acuerdo, se aprueba el proyecto. Este proceso coordina la cantidad de integrantes que tendrá el proyecto y asigna jefes para los equipos de trabajo, además gestiona los recursos necesarios para el desarrollo de los productos.

- **Proceso de Ejecución:** En esta etapa se elaboran los productos multimedia según los flujos de trabajo del paquete proceso de desarrollo.
- **Proceso de Control:** Este proceso supervisa y vigila todas las acciones realizadas, con el fin de asegurar que se cumple con las especificaciones y objetivos planificados. Esta tarea la realiza el grupo de calidad y el gerente de la factoría, quien se encarga de supervisar que los desarrolladores cumplen con las tareas planificadas, controlando el buen funcionamiento de la factoría en general.
- **Proceso de Cierre:** Una vez concluido el proyecto, este proceso se encarga de probar el producto antes de ser entregado y verificar que se cumplieron todos los requisitos iniciales. Después de esto se le entrega al cliente el producto final.
- **Proceso de Sostenibilidad:** Consiste en chequear el funcionamiento del producto para darle mantenimiento. Este proceso permite vigilar de manera sistemática si el producto funciona con éxito para evitar insatisfacciones por parte del cliente. Controla los distintos cambios que se pueden producir a lo largo de la ejecución del proyecto y aplica medidas para corregir los errores detectados durante la ejecución.

2.4.1.2 Gestión del alcance

Los procesos de gestión del alcance son necesarios para asegurar que en el proyecto se incluyen todas las actividades necesarias para realizar el mismo con el éxito requerido. Los procesos que definen la gestión del alcance se basan en la definición y el control de lo que debe desarrollarse en el proyecto. Para realizar una correcta gestión del alcance se realizan las siguientes etapas:

Iniciación: Informar a la factoría el surgimiento del nuevo proyecto. En esta etapa se parte de una descripción detallada de los productos a elaborar; está propiciado por el cliente como parte de su solicitud de negocio. Se determina un criterio de selección del proyecto, donde se resumen los factores que influyen en la decisión de iniciar el proyecto. Las técnicas que se utilizan para determinar el alcance son:

- Métodos de selección de proyectos: Este método compara las alternativas posibles dentro de las restricciones del proyecto y selecciona la vía más apropiada para desarrollarlo.
- Juicio de expertos: Determina los datos de partida para el proceso de selección, que pueden ser aportados por otras unidades organizativas.

En la etapa de planificación se obtiene un documento de lanzamiento del proyecto donde se reconoce formalmente la existencia del proyecto y los gastos necesarios para el comienzo del mismo.

Planificación del alcance: Describir el alcance del proyecto como base para decisiones futuras. Comprende la preparación de un documento con la definición del alcance de las actividades del proyecto. Se parte del documento de lanzamiento del proyecto y de las descripciones de los productos. Las técnicas que se siguen son:

- Análisis del proyecto: Se realiza un análisis detallado de los productos del proyecto, mediante técnicas de ingeniería de sistemas.
- Análisis costo – beneficios: Se estiman los costos tangibles e intangibles y los beneficios que reporta para verificar la rentabilidad.

Los productos que se obtienen en la planificación del alcance son la descripción del alcance que es un documento base que proporciona un entendimiento común del alcance del proyecto para todos los participantes. Además se genera un plan de gestión del alcance que describe cómo identificar y gestionar las variaciones del alcance para lograr su estabilidad.

Verificar el alcance: Obtener la aceptación del alcance por parte de los miembros del proyecto. Se toman como partida la descripción del alcance, las restricciones que tienen el proyecto y los resultados de otros procesos de gestión para estudiar e identificar los cambios del alcance del proyecto. Las técnicas y herramientas que se utilizan para determinar el alcance de un proyecto son las siguientes:

- Plantilla de estructuras de desagregación de proyectos: Define cada una de las actividades individuales que se realizan en el proyecto y la medida del avance de cada una de ellas.
- Descomposición: Subdivide los entregables del proyecto en componentes más pequeños y manejables.

Cuando se define el alcance del proyecto se obtiene la estructura de Desagregación del Proyecto. Es un documento que contiene una lista organizada de elementos que componen el alcance del proyecto. La estructura desagregada de proyecto se representa, en forma de árbol, hasta los niveles de detalle apropiados. Cada grupo de actividades tiene asignado un código identificador que facilite su integración con el sistema contable del proyecto y con el sistema de definición de tareas y lista de actividades.

Controlar los cambios del alcance: Identificar y vigilar el proceso de definición y aprobación de los cambios en el alcance durante el ciclo de vida del proyecto. Comprende el proceso de formalización de la aceptación del alcance del proyecto por los participantes, requiriendo la revisión de la estructura desagregada del proyecto y la descripción de los entregables, para asegurar que han sido definidos correctamente y satisfactoriamente. Se parte de los entregables del proyecto que muestran un resultado del trabajo realizado y los documentos que describen los productos a desarrollar en el proyecto como planificaciones, especificaciones y documentos técnicos. Se utiliza como técnica un proceso de inspección para evaluar y comprobar los resultados obtenidos y se obtiene como producto un documento de Aceptación Final del cliente.

2.4.1.3 Gestión del tiempo

La gestión del tiempo incluye los procesos necesarios para lograr la construcción del proyecto en el tiempo planificado. Para hacer una buena estimación del tiempo se debe definir claramente el objetivo del proyecto, determinar las actividades que se quieren llevar a cabo, un calendario de trabajo y fijar la duración y costo de cada actividad, así como el hito que marque el fin de una actividad y el comienzo de la otra. Para hacer una adecuada planificación de las tareas se propone trabajar con el Microsoft Project para la construcción del diagrama de Gantt. Se recomienda también elaborar un Diagrama Pert, éste es un conjunto de puntos (nodos) unidos por flechas que representan las relaciones entre las tareas del proyecto, su predecesor y el tiempo de duración de cada una.

Realizar la gestión del tiempo incluye las siguientes actividades:

- Identificar las actividades específicas del cronograma que deben ser realizadas para producir los diferentes productos entregables del proyecto.
- Identificar y documentar las dependencias entre las actividades del cronograma.
- Estimar el tipo y la cantidad de recursos necesarios para realizar cada actividad.
- Estimar la duración de las actividades que será necesario para desarrollar el proyecto.
- Desarrollar un cronograma que analice las secuencias de las actividades, la duración de las actividades, los requisitos de recursos y las restricciones para crear el proyecto.
- Controlar todos los cambios que se realicen en el cronograma del proyecto.

2.4.1.4 Gestión del costo

La gestión de costos del proyecto incluye todos aquellos procesos necesarios para asegurar que el proyecto se va a desarrollar conforme al presupuesto establecido. Se basa principalmente en los diversos gastos que se generan durante la realización del proyecto. Para elaborar la gestión del costo se desarrollan los siguientes procesos:

- Evaluación de recursos: Identificar los recursos y las cantidades a utilizar en el transcurso del proyecto. Para realizar la gestión de costos se parte de la estructura de desagregación y la definición del alcance del proyecto, además de tener un previo conocimiento de la disponibilidad de los recursos. Se necesita contar con personas capacitadas en economía para determinar el costo de estos productos y se obtiene un documento que resume las necesidades de recursos con la descripción de los mismos, la cantidad y el modo de obtención.
- Estimar los costos: Calcular el costo para cada recurso. Esta estimación puede estar sujeta a cambios ya que evalúa las diferentes alternativas del costo total del proyecto. Se toma como punto de partida la estructura de desagregación del proyecto, las necesidades de recursos y los costos unitarios de los recursos, además de una estimación del tiempo de duración de cada actividad. La herramienta que se propone para utilizar la estimación del costo es mediante el método COCOMO II, como jerarquía de modelos de estimación de costos que permite estimar el esfuerzo de desarrollo en personas-mes y está basada en líneas de código.
- Control de costos: Gestiona todas las desviaciones y cambios producidos en el presupuesto de los costos del proyecto. Estudia la forma de influir los factores que crean cambios en el presupuesto base del proyecto, para asegurar que los cambios sean beneficiosos. Para esto se mide constantemente el avance de los gastos incurridos para determinar la influencia y las desviaciones de los plazos estimados.

2.4.1.5 Gestión de la calidad

La gestión de la calidad contiene los procesos necesarios para asegurar que el proyecto satisface todas las necesidades previstas. Incluye las actividades de la gestión del proyecto que determinan la política de calidad mediante un plan de calidad. Los procesos que aquí se definen son los siguientes:

- Planificación de la calidad: Se identifica el estándar de calidad relevante para el proyecto y se especifica cómo implementarlo. Se determina utilizar el estándar ISO 9001 ya que este modelo comprueba la calidad en todas las etapas de desarrollo de un producto de software. La empresa debe tener un equipo de calidad y un jefe que responda por el funcionamiento de su equipo de

trabajo, además debe ser el responsable de la implantación de la norma ISO 9001 para determinar la calidad de los productos y CMMI para la mejora de procesos de la organización.

- Control de calidad: Se encarga de efectuar pruebas de calidad que verifiquen que el producto está listo para pasar a la siguiente fase.
- Aseguramiento de la calidad: Se evalúa de forma global los rendimientos del proyecto. El jefe de equipo debe conocer las expectativas del mercado y enfocar su política estratégica hacia el cliente. En esta fase todo el equipo de trabajo de calidad, está a cargo de comprobar que se cumple con los estándares requeridos y los aspectos necesarios para obtener un producto acorde con los requisitos especificados por el cliente.

2.4.1.6 Gestión de recursos

La gestión de los recursos describe los procesos necesarios para coordinar todo lo necesario para la puesta en marcha del proyecto. Aquí se planifica la cantidad de participantes que habrá y se seleccionan las personas adecuadas y necesarias para el desarrollo del mismo. Además se desarrollan las habilidades individuales para mejorar el rendimiento del proyecto.

En el modelo de factoría aplicando inteligencia la entidad Gestión de proyecto define la estructura organizacional que tendrán los trabajadores de la factoría. Según Pressman, “La mejor estructura de equipo depende del estilo de gestión de una organización, el número de personas que compondrá el equipo, sus niveles de preparación y la dificultad general de problema” (PRESSMAN 2002)

Pressman propone tres organigramas de equipos, el primero es el Descentralizado democrático (DD) es para proyectos que no tienen jefe y la comunicación entre los desarrolladores es horizontal. El segundo es Descentralizado controlado (DC), para proyectos con un jefe definido que coordina las tareas específicas a jefes secundarios los cuales dirigen pequeños grupos de trabajo. El tercero es el Centralizado controlado (CC) donde no existen equipos de trabajo y el jefe se encarga de resolver los problemas de alto nivel.

Tanto para definir los desarrolladores que integran el proyecto como para la selección del estilo se tuvieron en cuenta siete factores claves:

1. La dificultad del problema a resolver.
2. El tamaño del programa(s).
3. El tiempo de vida del proyecto
4. La calidad requerida y fiabilidad del sistema que se va a construir.

5. La rigidez de la fecha de entrega.
6. El grado de comunicación requerido para el proyecto.

Por estas razones y las características del proyecto se determina que la estructura de organización debe ser el Descentralizado controlado (DC). Cada equipo debe estar integrado por un grupo de 3 a 6 personas porque los productos son de corta duración y no tienen tanta complejidad. Con esta cantidad de desarrolladores por equipo se logra una comunicación más fluida y se obtienen productos con la calidad requerida en el tiempo estimado. Estos equipos variarán en dependencia de la carga de trabajo y debe existir al menos un equipo por cada flujo de trabajo que se desarrolle en el proyecto. Además, se debe gestionar y recolectar toda la información necesaria para la auto-preparación de los integrantes del proyecto, así como las formas de comunicación, ya sean reuniones, correos y citas. El documento “datos del equipo de trabajo” registra las personas involucradas en el proyecto y el rol que desempeña. (Ver Anexo 10).

2.4.1.7 Gestión de riesgos

La gestión de riesgos se encarga de identificar y analizar todos los riesgos que ponen en peligro el éxito de un proyecto y propone medidas para evitarlos o solucionarlos. Su principal objetivo es optimizar los efectos positivos para el proyecto minimizando los negativos. Gestionar los riesgos implica identificar, estudiar y eliminar las fuentes de riesgos antes que amenacen el éxito del proyecto. Permite conocer con anterioridad los posibles fallos que aparecerían y la forma de eliminarlos. La gestión de riesgos se controla mediante los siguientes niveles:

- Prevenir errores: Se detectan todos los posibles riesgos que puedan surgir a la hora de desplegar el proyecto.
- Mitigar errores: Una vez determinados cuales son los riesgos se busca la manera de solucionarlos. Se elabora un plan para cada riesgo de forma tal que lo corrija antes de que este llegue a convertirse en un problema.
- Corregir errores: Una vez que el riesgo se produce, debe ser detectado y corregido lo más rápido posible.

A continuación se presenta una lista de riesgos para el proyecto y la forma de mitigar cada uno de ellos, además de un plan de contingencia en caso de que el riesgo se convierta en un problema. Esta lista debe ser revisada por el gerente de la factoría para detectar si se encuentra en presencia de un riesgo, cuales son los planes de contingencia que se deben llevar a cabo y como prevenirlos.

Tabla 2.2 Plan de riesgos, mitigación y contingencia para el proyecto

Riesgos	Estrategia de mitigación	Plan de contingencia
1. Falta de entendimiento entre el gerente y los miembros de la factoría.	Hacer una reunión semanal con los desarrolladores para que el gerente les explique las tareas que cada uno debe desarrollar.	Asignar jefes de equipos capacitados para dirigir y orientar a los desarrolladores en ausencia del gerente de la factoría.
2. No existe un compromiso de los desarrolladores de realizar su trabajo con eficiencia y sistematicidad.	Explicarles a todos los miembros de la factoría las ventajas y el compromiso moral de su trabajo, para lograr una mayor motivación y compromiso laboral.	Analizar cada estudiante y profesor para ver los factores que impiden su cumplimiento y tomar medidas que lo comprometan a trabajar eficientemente.
3. Falta de información y conocimientos de los desarrolladores de las herramientas y tecnologías de trabajo.	Impartir cursos de capacitación que brinden los conocimientos necesarios para ejercer las tareas.	Buscar personas con más experiencia laboral para que apoyen el trabajo de los desarrolladores con menos experiencia y los guíen en sus funciones.
4. Ausencia o carencia de recursos.	Buscar más desarrolladores capacitados y asignar más máquinas o trabajar con la misma cantidad de personas pero con más horas de trabajo.	Hacer una replanificación sobre la cantidad de recursos materiales y humanos que se dispone, para que los productos se entreguen en los plazos planificados.
5. Inexperiencia del trabajo en equipo.	Se imparte un curso intensivo sobre la herramienta TSP y se designa un jefe con conocimientos del tema para que oriente y guíe al equipo de	Organizar reuniones con el gerente de la factoría para corregir los errores y brindarles una mejor orientación a los equipos de trabajo.

	desarrollo.	
6. Planificaciones irreales.	Se debe hacer una planificación sobre la base de los recursos disponibles y con las herramientas que propone el modelo de factoría.	Se deben hacer planificaciones con varias semanas de más, para prever informalidades en la entrega.
7. Los desarrolladores tienen otras tareas importantes que realizar ya que son estudiantes y trabajadores.	Disponer de 4 horas diarias de trabajo y estimar planificaciones personales para cumplir con las funciones del proyecto.	Tener varios desarrolladores de reserva con los conocimientos necesarios para en caso de atrasos acudir a estos, para que se entreguen los productos en el tiempo planificado y con la calidad requerida.
8. Atrasos o errores en el diseño.	Debe existir una persona en el proyecto capacitada para revisar los diseños antes de comenzarse a elaborar los productos multimedia.	Contar con un equipo altamente capacitado para si se entrega un producto con errores o retrasos, estos se puedan corregir en el menor tiempo posible antes de afectar el tiempo límite de entrega de los productos.

2.4.2 Proceso de desarrollo

El proceso en el desarrollo de un software define todas las actividades necesarias para convertir los requisitos del usuario en un producto final. Como los productos multimedia a elaborar son relativamente sencillos y se comienzan a desarrollar a partir de la fase de elaboración, se tomó la metodología RUP que tiene la ventaja de adaptarse a las características y tamaño del proyecto. A continuación se muestra el proceso de desarrollo de los productos para la Factoría de Software, los flujos identificados, roles, actividades y artefactos necesarios.

2.4.2.1 Elaboración

En el proceso de desarrollo de la factoría, el ciclo de desarrollo de un producto multimedia en el proyecto necesita de dos artefactos fundamentales: el guión de contenidos entregado por la empresa cliente y las diagramaciones del equipo de diseñadores gráficos.

El guión de contenidos tiene la información sobre el producto a implementar, donde se especifican los requerimientos y especificación que debe tener el producto; este guión posee un mapa de navegación preliminar, los módulos que tendrá el producto y un formato de guión técnico con una explicación detallada de cada pantalla tanto de su contenido como de la interacción del usuario final con el sistema. Además, brinda información del tema sobre el cual se va elaborar el producto multimedia y es la base para construir las diagramaciones, estas por su parte son un prototipo de cómo debe quedar el sistema incluyendo principios de diseño gráfico, interactividad y contenido.

En muchos de los casos las diagramaciones no concuerdan con el guión de contenidos, trayendo consigo que los desarrolladores tengan que dedicar tiempo a buscar la forma de entender la información del guión y adaptarla al diseño entregado, provocando descensos en la productividad, pérdida de tiempo y errores en el producto. Debido a esto, se debe rediseñar el guión de contenidos con las diagramaciones aprobadas por el cliente, de modo que cumpla con los requisitos. También como cada portal incluye una determinada cantidad de contenidos a desarrollar que tienen características muy similares, un cambio en el diseño del portal implica que se tengan que cambiar todos sus contenidos. Por lo tanto, se hace necesario la definición de un estándar para cada portal, que permita la reutilización de los contenidos.

El flujo de sucesos se basa en entregar las diagramaciones, verificar que cumplen con los requisitos planteados en el guión de contenidos y en caso contrario ser devueltas a los diseñadores. Luego el arquitecto comienza a elaborar la plantilla del portal, garantizando que esta se rija por las especificaciones del guión rediseñado. Esto facilitará el proceso de montaje de las medias a los contenidos, mediante la transformación de los requerimientos en los esqueletos de los contenidos a montar. La salida de esta etapa es la plantilla que regirá el formato de los contenidos del portal.

Simultáneamente el resto de los desarrolladores construyen las actividades interactivas, utilizando la herramienta Flash para ganar en tiempo, luego se identifican los posibles componentes a reutilizar y todos se almacenan en el repositorio de componentes para su posterior reutilización en otros portales Web.

Una vez elaboradas las plantillas y las actividades interactivas se realiza el montaje de las medias a cada contenido del portal que entrega el equipo de diseñadores. Si el arquitecto considera necesario incluir componentes del repositorio se los solicita al administrador del repositorio, colocándolo a su vez en la arquitectura. Seguidamente el gerente de la factoría verifica el diseño de la plantilla y comienza a planificar

el montaje de las medias a los contenidos en el plan de montaje, distribuyendo el contenido a los líderes de cada equipo de desarrollo para que estos asignen las tareas específicas a los programadores. (Ver Anexo 11)

Los programadores deben implementar, probar y controlar la calidad de los contenidos en que trabajan, y terminando la tarea asignada tienen que solicitar una revisión al gerente de desarrollo y al encargado de calidad. Comprobada la calidad se aprueban los contenidos montados para que pasen a integrarse al resto de los contenidos del portal. Finalmente el arquitecto es el responsable de la integración final, junto a otros programadores si es necesario. Durante toda esta etapa de construcción se controla la calidad mediante métodos de prueba que verifiquen el funcionamiento íntegro de cada producto. En el anexo 12 se muestra la tabla descriptiva de este flujo de procesos.

2.4.2.2 Prueba

La cantidad de pruebas que se le apliquen a un producto garantizan su calidad, por lo que para reducir los riesgos de posibles fallos en los contenidos o en la programación de las actividades interactivas se necesitan realizar pruebas minuciosas y bien planificadas. Cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de pruebas y la depuración y evaluación de los datos resultantes.

La etapa de prueba comienza cuando llega al equipo de prueba el guión rediseñado, la plantilla de diseño y el resultado del montaje de los contenidos para probar. Estos artefactos son imprescindibles para que el jefe de pruebas inicie la planificación de las mismas con la elaboración del Plan de prueba. Luego, pasa al diseño de las pruebas, especificando los casos de prueba y los procedimientos para ejecutarlas. Estos elementos son los que utilizan los probadores para ejecutar las pruebas, detectando errores que puedan existir. El jefe de pruebas registra los errores, emite un reporte de defectos y los envía a flujos de trabajo anteriores para que sean corregidos y una vez rectificadas se vuelven a enviar con el documento de errores corregidos. El jefe de pruebas debe evaluar la prueba, comparando los resultados con el plan que se tenía elaborado inicialmente. La documentación de la evaluación de estos resultados sirve para futuras pruebas. (Ver Anexos 13 y 14)

Las otras pruebas que se deben realizar en esta etapa son las pruebas de unidad, enfocadas al código fuente de los componentes para verificar todos los flujos de control. Dentro de las pruebas de unidad existen varios métodos, sin embargo debido a los productos que se desarrollan en el proyecto se determina realizar pruebas de caja blanca para verificar el código programado y pruebas de caja negra

para supervisar que la interfaz gráfica está acorde con el guión técnico y el informe preliminar. Se realizan pruebas de sistema para probar el software funcionando como un todo y finalmente una prueba de aceptación antes del despliegue final del sistema.

2.4.2.3 Entrega

Esta etapa comienza al terminar las pruebas del producto. El gerente de la factoría es el responsable de entregar la versión del producto al cliente y este por su parte también comprueba la calidad de la entrega y determina la aceptación de la versión del producto o puede solicitar una nueva versión.

Las versiones del producto deben ser almacenadas en el repositorio de componentes de la factoría, así el administrador del repositorio es el encargado de mantener actualizada toda la información. El proceso para almacenar un producto en el repositorio comienza por la solicitud de almacenamiento y seguidamente el administrador lo documenta junto al gerente de la factoría. (Ver Anexos 15 y 16)

2.4.3 Centro de Inteligencia

Este paquete se debe desarrollar con mayor profundidad en una segunda fase de la factoría, su objetivo es la orientación estratégica para la toma de decisiones internas o externas a la factoría. El modelo de factoría aplicando inteligencia define la entidad Centro de Inteligencia como 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)

Estos aspectos se resumen en dos grupos de investigación: interna y externa. En el proyecto debe existir un gestor de información interna y un gestor de información externa, para cada grupo de investigación respectivamente. Estos trabajadores son los encargadas de llevar a cabo la orientación estratégica para la toma de decisiones a todos los miembros de la factoría y un tercer trabajador en el rol de gestor empresarial, con conocimientos empresariales para la toma de decisiones a ese nivel, además analizar el mercado exterior para valorar los costos de todos los recursos de la factoría, así como la de los productos que se elaboren. El gestor empresarial también está a cargo de la inteligencia empresarial, reuniendo todos los datos obtenidos por el responsable de la gestión interna y externa y en base a esto tomar las decisiones imprescindibles para el funcionamiento íntegro y correcto de la factoría.

La gestión del conocimiento debe ser desarrollada por el gestor de información interna, éste es el encargado de promover el conocimiento para todos los miembros de la organización. Además, tiene la responsabilidad de almacenar y mantener actualizados los documentos de información generados en la factoría y relacionados con el conocimiento que se maneja. Debe buscar las vías para la capacitación de los trabajadores de la factoría y garantizar su seguimiento.

El gestor de la información interna debe comprobar que cada vez que se incorpore un nuevo integrante a la factoría, este cuenta con los conocimientos y habilidades necesarios para el uso de las herramientas, lenguajes de programación y todas las normas establecidas en el proceso, previniendo con esto riesgos de inconsistencias en los tiempos de entrega de los productos.

La vigilancia tecnológica la realiza el gestor de información externa, ya que se encarga de investigar cómo se encuentra el mundo exterior a la factoría en todo lo concerniente a tecnología. Esta persona investiga sobre los nuevos y mejores lenguajes de programación, plataformas y sistemas operativos sobre los que trabajan estos programas, así como los componentes de hardware que sean de utilidad en el manejo de las multimedia. También se puede promover la investigación en función de las necesidades y demandas del cliente.

Todos estos trabajadores desarrollan la prospectiva cada uno velando por sus responsabilidades, ya que miran el futuro con sus riesgos y ventajas desde la situación actual de la factoría.

2.4.3.1 Capacitación

Con el objetivo de capacitar a los integrantes del proyecto para la transferencia del modelo de la factoría se debe realizar un taller para introducirlos y concientizarlos en la importancia de aplicar este enfoque. Los temas a debatir abarcan la familiarización con los antecedentes y definiciones de factoría, los modelos que existen, principalmente el que se desea transferir, sus características, quienes los aplican y que beneficios ha tenido con su aplicación.

La línea de producción debe garantizar que se les impartan cursos a sus trabajadores para que tengan dominio de los conocimientos necesarios para un buen desempeño laboral en la factoría y evaluarlos antes de comenzar la producción, preparándolos con proyectos ficticios pero con características muy similares, evitando así falsas suposiciones que pueden traer inconsistencias en los tiempos de entrega, por diferencias en la velocidad del aprendizaje. Cada vez que a la factoría se incorpore un nuevo integrante, deberá estar ya relacionado con el uso de las herramientas y las normas establecidas en el proceso.

La plantilla de solicitud de cursos de capacitación confeccionada por el gerente de la factoría puede observarse en el anexo 17, contiene los lenguajes principales que los miembros de la factoría deben dominar: HTML, CSS, JavaScript y opcionalmente XML para tareas posteriores.

Los trabajadores de la factoría como condición inicial para formar parte del proyecto deben tener conocimientos previos en el trabajo con las herramientas de autor Dreamweaver y Flash, ya sea que por haber cursado alguno de los cursos optativos relativos o por exámenes de suficiencia.

Para la utilización de TSP, se diseñan cursos cortos de cuatro frecuencias presenciales para debatir cada uno de los roles a través de casos de estudio. El nivel de conocimiento que se adquiere en el curso está fuertemente relacionado con la investigación y trabajo individual de los miembros del curso.

Para disciplinar a los trabajadores con el uso de PSP se realiza a modo de taller, donde se les explica las principales herramientas que deben utilizar. En el taller se les facilita un Cuaderno del Ingeniero en formato digital, creado con el Microsoft Excel, que contiene las plantillas principales para trabajar, como son: Cuaderno de Registro de Tiempos, Resumen Semanal de Actividades y Cuaderno de Defectos.

2.4.4 Repositorio

En el enfoque de Factorías de Software la reutilización juega un papel fundamental en el proceso de desarrollo, dado el aumento en la productividad, la reducción del riesgo y el incremento de la calidad.

El administrador del repositorio debe configurar el repositorio para insertar, modificar o eliminar componentes, limitando el acceso a los desarrolladores en dependencia del rol que desempeña en la factoría. Su misión es mantener actualizado el repositorio para facilitar la reutilización eficaz de sus componentes.

El repositorio debe contar con un catálogo donde se recojan los componentes reutilizables. La información que debe tener cada componente para facilitar su búsqueda es su nombre, palabras claves, grupo al que pertenecen y el tipo de componentes, además de registrar la fecha de creación y donde se utilizan. (Ver anexo 18)

Los componentes se dividen en dos grandes grupos: Componentes de código y Activos del proceso. Los componentes de código pueden ser: una clase, un procedimiento o función, un módulo, un subsistema, una aplicación. Los activos del proceso pueden ser: patrones de diseño, algoritmos, medias, diagramaciones, guiones, manuales y documentación o un modelo.

La estructura que tiene la información del proyecto, almacenada en el repositorio, debe estar organizada por portales Web, cada uno con sus pantallas, medias, guiones técnicos y las diagramaciones. Cada desarrollador puede acceder al servidor y tomar las medias que necesita para elaborar sus productos y los almacena de nuevo en el mismo para su posterior reutilización. La estructura de la información está establecida en el Anexo 19. También se debe guardar el resto de la información generada durante el proyecto.

El administrador debe realizar salvadas diarias de su repositorio en el repositorio central a nivel de universidad para la línea de software educativo, para evitar problemas de conexión o pérdidas.

2.4.5 Personas

El factor humano es de vital importancia para el desarrollo de cualquier Factoría de Software, por lo que se debe tener bien definido y organizado por roles todo el personal que allí labore y motivarlos a desarrollar un proceso con calidad a través de métricas que ayuden a la mejora continua del mismo.

2.4.5.1 Roles de la factoría

Jacobson plantea que “un proceso define “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo. ”(IVAR JACOBSON 2000) En otras palabras, los trabajadores realizan las actividades en el proceso de desarrollo de software que generan artefactos por cada flujo de trabajo, responden al “quién” dentro de un proceso de desarrollo de software, por lo que para la Factoría de Software propuesta se observa en la figura siguiente la estructura organizativa de los roles y su correspondiente descripción.

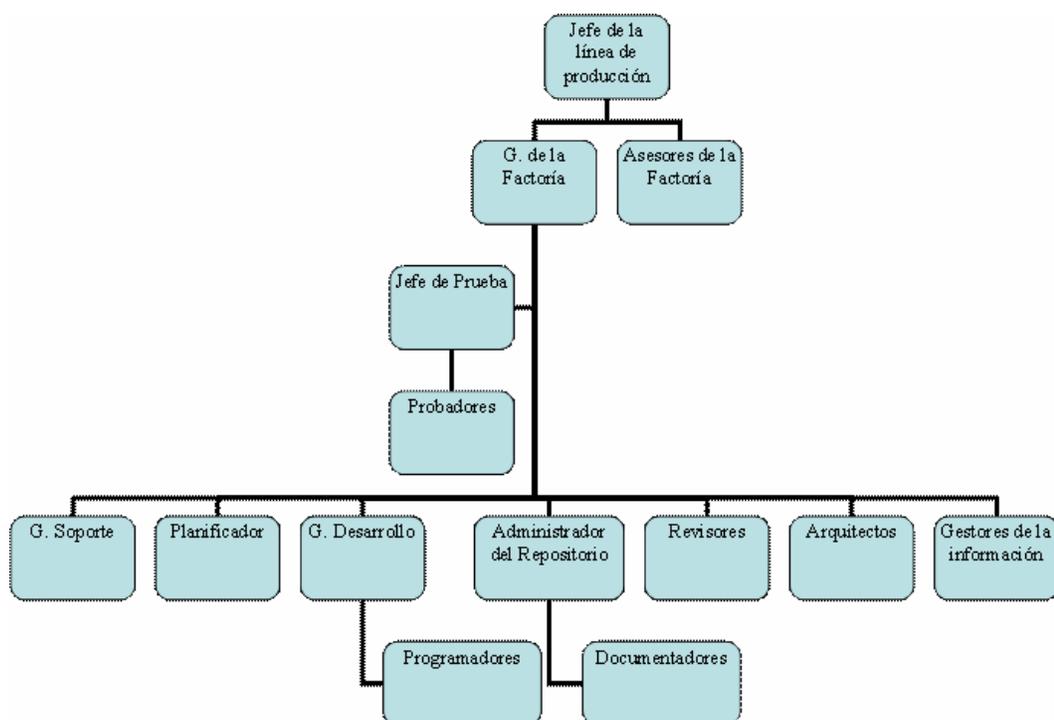


Figura 2. 1 Estructura Organizacional

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 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: es el encargado de reunirse con los clientes, asignar los recursos necesarios para la producción y contactar con el cliente para las entregas de cada versión realizada.

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 y participa con el gerente en las reuniones con los clientes. Es el responsable de velar por el cumplimiento de los acuerdos tomados y revisar el plan de trabajo de cada miembro de la factoría para tomar acuerdos y medidas.

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: guían a cada equipo de desarrollo y asignan 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.

Revisores: revisan los guiones de contenidos y las diagramaciones para cada producto multimedia para documentarlos y corregirles los errores que tengan.

Arquitectos: diseñan las plantillas de contenidos e identifican los componentes reutilizables. Además integran los contenidos al portal Web.

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 interna: promueven el conocimiento para todos los miembros de la organización. Gestionan los recursos técnicos y humanos para los cursos de capacitación. Almacenan y actualizan los documentos de información; además de comprobar los conocimientos y habilidades adquiridos por los trabajadores de la factoría.

Gestor de información externa: investiga el mundo exterior a la factoría en cuanto a la tecnología. Promueve la investigación en función de las necesidades y demandas del cliente. Emite un reporte de la información externa para la toma de decisiones de la empresa y estimula la realización de trabajos investigativos.

Gestor empresarial: reúne los datos de la gestión interna y externa para la toma de decisiones. Agrupa todas las decisiones tomadas y emite un reporte de la información obtenida. Analiza el mercado exterior para valorar los costos de todos los recursos y productos.

Para un mejor entendimiento de los roles descritos, en el anexo 20 se muestra una definición de cada rol con sus responsabilidades, habilidades, conocimientos y valores.

2.4.5.2 Instrumentos PSP para utilizar

La entidad Personas del modelo de factoría aplicando inteligencia propone que los trabajadores implicados utilicen PSP. Los instrumentos de PSP pueden ayudar en gran magnitud al trabajo organizativo de cada individuo en la factoría; teniendo en cuenta que el personal de la misma es joven e inexperto en desarrollo de software, además de tener otras responsabilidades adicionales al desarrollo de software. Algunos de estos instrumentos son:

- *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 21).
- *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 22).
- *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 23).
- *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 24).

2.4.6 Bases Tecnológicas

La tecnología se muestra como un objeto imprescindible para el desarrollo de la sociedad, sus beneficios son incalculables y modifican el entorno en que habita el hombre actual. En la organización de una Factoría de Software debe conocerse fundamentalmente la utilidad que presenta cada herramienta y las normas disciplinarias especificadas por los directivos que se deben cumplir.

2.4.6.1 Tecnologías y herramientas

La Factoría de Software debe identificar primeramente la plataforma sobre la que va a desarrollar su línea de producción y el grupo de herramientas a utilizar. Las herramientas en esta línea de producción son:

Dreamweaver / IDE: Esta herramienta de uso sencillo proporciona desarrollo robusto de aplicaciones, creación visual y rápida de prototipos, depuración integrada. Es la principal herramienta a utilizar para montar los contenidos educativos. Exige un procesador Intel Pentium III de 600 MHz o equivalente, 256 MB de RAM como mínimo y 275 MB de espacio disponible en disco.

Flash MX 2004: Esta herramienta permite, además de la creación visual y rápida de prototipos, la programación orientada a objetos y depuración integrada. Es muy utilizada para diseñar actividades interactivas, por lo que su uso está basado en la creación de juegos para los contenidos educativos. Exige del hardware las mismas potencialidades de la herramienta anterior.

Microsoft Project: Es una aplicación que ayuda al usuario a crear planes de proyectos, comunicarlos a otros usuarios y adaptarse a los cambios a medida que éstos se van produciendo. Se utiliza por los directivos de la factoría para introducir las tareas del proyecto y sus duraciones. La organización de tareas es en estructura jerárquica y su vinculación para diferentes tipos de relaciones está sumamente simplificada. Con un procesador Pentium II de 200 Mhz. y 32 MB de memoria RAM, el programa se desenvuelve normalmente.

TortoiseSVN: Es una herramienta de configuración muy popular, ya que es un cliente gratuito de código abierto para el sistema de control de versiones Subversion, maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central, lo que permite que pueda recuperar

versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. Es compatible con Windows 2000 SP2, Windows XP o superiores. Su elección también depende del esquema de configuración que tiene diseñado actualmente la Dirección de Software Educativo de la universidad, la experiencia que existe en su uso y su capacidad de adquisición.

En el anexo 25 aparece un resumen de cada uno de las herramientas y su descripción, las cuales se deben instalar en todas las estaciones de trabajo y verificar su correcto funcionamiento. Todos los trabajadores de la factoría necesitan utilizarlas para la producción. También es imprescindible contar con un servidor local dentro del centro de trabajo con la información actualizada del servidor central para prevenir pérdidas de información o fallos de conexión. Los instaladores deben ser almacenados en el repositorio para evitar daños y el gerente de soporte es el encargado de mantener actualizadas las herramientas.

2.4.6.2 Mecanismos

Las listas de chequeo constituyen un excelente mecanismo para el control de la calidad. Los productos terminados deben cumplir las especificaciones de calidad, por lo que las listas de chequeo se definen para cada fin de etapas y para cada revisión. En todas las fases del desarrollo se debe evaluar el cumplimiento de estas listas para hacer el menor número de correcciones posibles y garantizar que el producto que se elabora en cada fase tenga la calidad esperada.

Las políticas de seguridad de la información (PSI) pertenecen a los mecanismos de seguridad de la información. Estas políticas describen accesos como: la información básica, software que se deben instalar, garantía del producto en uso, restricciones sobre los dispositivos de acceso a la estación de trabajo y del personal ajeno a la factoría, incluyendo medidas que se deben tomar para traspasar información a través de Internet. Estas se pueden observar en el anexo 26.

2.5 Implantación del proceso de transferencia

Una vez definida la estrategia para realizar la transferencia, es necesario reunirse con todos los miembros de la misma para explicarles en qué consiste y las ventajas que ofrece este cambio de tecnología. El grado de motivación de los trabajadores es vital para el éxito de la estrategia.

Después de esto se gestionan todas las herramientas y recursos financieros, humanos y tecnológicos para comenzar la implementación de la estrategia de transferencia del modelo de Factoría de Software.

Luego de la puesta en marcha de la factoría, se debe realizar un seguimiento continuo del funcionamiento y desempeño laboral de cada uno de sus miembros.

Existen una serie de factores que ponen en riesgo el funcionamiento exitoso de la estrategia propuesta. Por lo que se determinó planificar, identificar y analizar todos aquellos posibles aspectos que de una manera u otra pudieran impedir que la estrategia de transferencia del modelo de factoría cumpla sus objetivos iniciales. Además, se determinaron opciones preventivas y acciones a tomar por si los riesgos afectaban la transferencia. En la siguiente tabla se muestran estos riesgos, su plan de mitigación y contingencia.

Tabla 2.3 Plan de riesgos, mitigación y contingencia para la implantación de la factoría

Riesgos	Estrategia de mitigación	Plan de contingencia
1. Falta de entendimiento entre los asesores de la factoría y el gerente.	Hacer una reunión para explicarle al gerente de la factoría en que consiste y que se pretende lograr con la estrategia de transferencia.	Reunirse con el gerente de la factoría cada 15 días para ver su funcionamiento. Debe existir otra persona preparada para apoyarlo, en caso de que éste no tenga los conocimientos necesarios para dirigirla.
2. No existe un compromiso con los máximos responsables de la factoría del interés personal de su funcionamiento exitoso.	Explicarle a todos los integrantes del proyecto las ventajas que brinda aplicar el modelo de Factoría de Software propuesto, para lograr una mayor motivación y compromiso de trabajo.	Analizar cada estudiante y profesor para ver los factores que impiden su cumplimiento y tomar medidas que lo comprometan a trabajar eficientemente.
3. Falta de información y conocimientos de los desarrolladores del modelo de factoría que se propone.	Los asesores de la factoría deben reunirse con los desarrolladores y explicarles en que consiste la Factoría de Software y que es lo	Buscar personas con más experiencia laboral para que apoyen el trabajo de los desarrolladores con menos

- | | | |
|--|---|---|
| | que cada uno debe hacer. | experiencia. |
| 4. Mal diseño de la factoría o de su estrategia de transferencia. | Verificar la estrategia propuesta antes de su implantación con personas expertas en la materia. | Si existe algún error en el diseño de la factoría o en su estrategia de transferencia, ésta debe ser rediseñada, analizando los factores de riesgo. |
| 5. El proyecto para el cual se propone la factoría es muy pequeño y no cumple con todos los requisitos descritos en el modelo. | Implantar solo una parte de la factoría, la que se ajuste con las necesidades del proyecto. | Simplificar más el modelo propuesto hasta que cumpla con los requisitos y características del proyecto. |
| 6. No se cuenta con la cantidad de personas suficientes para distribuir cada rol propuesto en el modelo. | La factoría se adecua tanto a nivel empresarial como a nivel de proyecto. Por lo que solo se definirán los roles necesarios para el proyecto. | Analizar las funciones de cada desarrollador y asignarle el rol que solo cumpla con sus funcionalidades. |
| 7. Desconocimiento por parte de los desarrolladores de la tecnología y metodología propuesta. | Impartir cursos a todos los desarrolladores y al líder del proyecto de las herramientas y tecnologías que se proponen utilizar. | Hacerles una evaluación para verificar el dominio de las herramientas y vincular los desarrolladores para que compartan su experiencia. |
| 8. La universidad no cuenta con muchas personas expertas en el tema de Factoría de Software, por lo que es una temática nueva y sin precedentes. | Los asesores de la factoría deben hacer revisiones bibliográficas sobre otros modelos implantados en el mundo para obtener cierta experiencia. Además de la asesoría de personas expertas en dirección empresarial. | Crear un centro de inteligencia que reúna a los expertos y prepare a los coordinadores de cada factoría. |

Estos riesgos dan un margen para que la estrategia de transferencia del modelo de factoría propuesto para el proyecto no se efectúe exitosamente. No basta con llevar a cabo la estrategia propuesta de manera rigurosa, lo más importante para que obtenga los beneficios esperados es el interés personal de cada uno de los miembros del proyecto, principalmente de las personas encargadas de implantar la factoría y los que la dirigen. Por eso, se debe hacer un seguimiento minucioso a cada uno de los riesgos definidos con anterioridad para obtener los resultados esperados.

2.6 Conclusiones

En este capítulo se describió como se realiza el proceso de desarrollo de software educativo en la UCI, donde se observan deficiencias que impiden dar una respuesta rápida y eficiente a sus clientes. Se describieron los principales métodos, procedimientos y técnicas utilizadas para la investigación. Además, se definió la estrategia de transferencia del modelo Factoría de Software aplicando inteligencia a un proyecto para contenidos educativos con herramientas de desarrollo Web y los riesgos que pueden surgir durante la implantación de la factoría.

CAPÍTULO 3: RESULTADOS

3.1 Introducción

Este capítulo muestra los resultados obtenidos después de probar algunos de los elementos propuestos en la estrategia de transferencia del modelo de Factoría de Software aplicando inteligencia al proyecto CNTI – Contenidos Educativos.

Parte de la estrategia definida se probó durante un mes al proyecto productivo de la UCI que responde a la elaboración de contenidos educativos para Venezuela. Aquí se resume la experiencia adquirida y una demostración de las ventajas que puede ofrecer aplicar el enfoque Factoría de Software.

3.2 Resultados de las técnicas utilizadas

Se aplicaron las encuestas a los 25 desarrolladores previstos inicialmente. Con esto se llegó a la conclusión de que el 80% de los encuestados no sabía en qué consistía su proyecto y qué objetivos tenía. El 76% no tenía conocimientos del rol que desempeñaba en el proyecto y solo un 12 % conocía de planificación, control de actividades y del trabajo personal y en equipo utilizando PSP y TSP respectivamente.

Solo un 12% sabía de la existencia de alguna metodología para el trabajo con multimedia. El 88 % de los encuestados determinó que la comunicación interpersonal era mala y en algunos casos regular, además existía un 92% que desconocía totalmente las herramientas a utilizar en el proyecto. (Ver anexo 27)

Las comunicaciones con el líder del proyecto permitieron conocer que los roles no estaban debidamente establecidos ni definidas las actividades que tenían que realizar. Las estimaciones no estuvieron basadas en la capacidad real ni en el tiempo de trabajo disponible por cada miembro del proyecto, sino que se definieron en función de una situación ideal: buena preparación de los miembros del proyecto, las entregas de diseño (informes de diagramación) acorde con lo que plantean los guiones y contando con todas las medias pautadas en los guiones. Tampoco tenían definida políticas de seguridad de la información dentro del proyecto. (H. Ordaz, comunicación personal, 24 de abril de 2007)¹

¹ Ing. Harold Ordaz Valdes, Líder del proyecto CNTI-Contenidos Educativos, Universidad de las Ciencias Informáticas, Ciudad de la Habana, Cuba, 2007. (Comunicación personal)

3.4 Observaciones sobre la prueba de la estrategia

La estrategia contiene una serie de funcionalidades que no se pudieron probar en el proyecto, debido a que esta se terminó de definir cuando el proyecto llevaba funcionando cerca de 3 meses. Esto impidió determinar un estudio de factibilidad, además de la gestión inicial del alcance, tiempo y otras funcionalidades que se proponen en la Factoría de Software para optimizar la producción.

Los primeros pasos de la estrategia están implícitamente desarrollados en el documento, cada uno de los resultados obtenidos se describirán en el mismo orden en que se definieron y posteriormente el reporte a través de los resultados de las encuestas realizadas al finalizar la prueba piloto.

3.4.1 Gestión de Proyecto

Este paquete incluye aspectos fundamentales para el éxito de un proyecto, como son definición del alcance, tiempo, calidad, costo, riesgos y recursos. Sin embargo de este paquete no se probó todo, por lo que solo se explicarán con mayor detalle aquellos donde se pudieron obtener resultados.

3.4.1.1 Gestión del alcance

El alcance del proyecto no fue determinado según lo planteado por la estrategia, ya que las herramientas que se proponen deben construirse durante la concepción del proyecto, o sea en sus inicios y la factoría se puso a funcionar cuando el proyecto tenía cierto tiempo de creado.

3.4.1.2 Gestión del tiempo

La gestión del tiempo no se probó en el proyecto, sino que el tiempo de desarrollo se definió de común acuerdo entre el cliente, el líder del proyecto y el representante de los diseñadores, fundamentalmente en función del cronograma de entregas de medias por los diseñadores gráficos.

Hay que tener en cuenta a la hora definir la gestión del tiempo la fuerza de trabajo. En este caso son estudiantes que solo pueden dedicarle 4 horas diarias a la producción, valorar la cantidad de máquinas disponibles para definir los horarios de trabajo y principalmente la experiencia del líder para asumir la responsabilidad de la dirección del proyecto y distribuir eficientemente las tareas.

3.4.1.3 Gestión del costo

El costo del proyecto fue otro de los aspectos que no se probó, debido a que estas estimaciones se establecen a un nivel más alto para determinar un precio al cliente sobre los productos de software que se elaboran y no están dentro del alcance de la factoría.

3.4.1.4 Gestión de la calidad

La gestión de la calidad implica hacer una planificación, control y evaluación que no se hace directamente en el proyecto. Sin embargo el equipo encargado de revisar la calidad debe guiarse por el estándar ISO 9001 para los productos y CMMI para los procesos dentro de la factoría.

3.4.1.5 Gestión de los recursos

En la gestión de recursos se determinó la cantidad de personas que trabajaron en el proyecto. La capacidad se redujo a 22 desarrolladores y se determinó utilizar como organigrama de equipos el descentralizado controlado (DC), por lo que se formaron 4 equipos de 4 integrantes con un líder o gerente de desarrollo para montar las medias, un equipo para las actividades interactivas, otro encargado de definir la arquitectura y el de investigación, ya que de esta forma fue más fácil controlar las tareas y el avance de cada trabajador, además para que todos tuvieran la oportunidad de dar nuevas ideas y propuestas de soluciones durante el desarrollo del trabajo en los equipos. Mediante las reuniones realizadas semanalmente cada desarrollador pudo dar las sugerencias pertinentes y plantear las dudas que no pudo solucionar dentro de su equipo de desarrollo.

Los datos de contacto de cada uno de los desarrolladores se registraron inicialmente en un documento en Excel, utilizando listas para facilitar la búsqueda por rol, estación de trabajo, grupo, etc, en caso de que alguno de los líderes los necesitara de carácter urgente.

3.4.1.6 Gestión de riesgos

En el capítulo dos se hizo un análisis de los riesgos que afectarían el proyecto y las posibles formas de enfrentarse a estas situaciones. A continuación se muestran estos riesgos y la forma en que incidieron sobre el proyecto CNTI – Contenidos Educativos.

- R1: Este riesgo no se dio ya que se previó la falta de entendimiento entre el líder del proyecto y los desarrolladores mediante las reuniones coordinadas semanalmente.

- R2: Este riesgo no se dio porque en la reunión inicial se le informó a cada desarrollador la importancia de su proyecto tanto para la universidad como para su experiencia laboral, así todos tomaron conciencia y valoraron sus actividades para realizarlas en tiempo y con la calidad exigida.
- R3: Este riesgo se dio porque los trabajadores no tenían dominio de las herramientas de trabajo y los cursos de capacitación que se planificaron no se terminaron de impartir, por lo que hubo que buscar personas con experiencia para que les enseñaran los conocimientos básicos para trabajar con ellas.
- R4: Este riesgo se dio porque no se pudo contar con más máquinas de trabajo, solo se dispuso de 10 máquinas para 22 desarrolladores. Por lo tanto se tuvo que acudir al plan de contingencia donde se hizo una replanificación del tiempo de máquina disponible, asignando horas extras de trabajo a cada trabajador.
- R5: La inexperiencia del trabajo en equipo fue algo que afectó al proyecto ya que nunca habían estado en un proyecto y no conocían las funciones que debía desempeñar cada rol. Para disminuir el impacto, el líder o gerente de la factoría se reunió con ellos y les explicó las tareas principales para motivarlos a que aumentaran su conocimiento investigando sobre el tema.
- R6: Para evitar este riesgo se designó un planificador en el proyecto que estimó el tiempo de desarrollo de cada actividad sobre la base de los recursos y herramientas disponibles, además cada gerente de desarrollo tuvo la responsabilidad de verificar el cumplimiento de la planificación del trabajo personal.
- R7: Este riesgo se dio ya que a pesar de disponer de 4 horas diarias de trabajo, se tuvo que acudir a desarrolladores de reserva para cumplir con la entrega en el tiempo planificado.
- R8: Este riesgo afectó al proyecto y se determinaron personas en el proyecto encargadas de verificar los diseños hechos por el grupo de diseñadores gráficos, sin embargo esto no se cumplió completamente.

3.4.2 Proceso de desarrollo

El paquete proceso de desarrollo no se pudo probar en la factoría. Las alternativas de solución a las deficiencias en la producción vinieron de niveles superiores. Es importante cuando se decida probarlo, que cada desarrollador en la etapa en que se encuentre, utilice las listas de comprobación propuestas en la Entidad Bases Tecnológicas para disminuir los errores y garantizar la mejora del proceso. Además, deben encontrar la manera de acercarse a cumplir los requerimientos de los estándares propuestos.

3.4.3 Centro de Inteligencia

De este paquete solo se probó una parte del proceso de capacitación puesto que el mismo se realizó desde los inicios del proyecto, antes de implantar la factoría.

Para pertenecer al proyecto se necesitaba tener dominio de los lenguajes de programación y plataformas de desarrollo en las que se iban a construir los productos, condiciones que no se validaron con anterioridad. La estrategia determina que se debe comprobar los conocimientos de cada nuevo trabajador, si se va a aplicar en un proyecto que comienza se debe primero hacer ejercicios pilotos muy parecidos a los que se enfrentarán después y así garantizar que todos tienen conocimiento de lo que deben hacer.

De los cursos solicitados antes de iniciar el proyecto solo se impartió el curso de HTML, provocando que los trabajadores no tuvieran suficiente conocimiento para enfrentarse a las tareas, a pesar de que las tecnologías propuestas forman parte de los cursos optativos que brinda la facultad.

Para comenzar a probar la estrategia se realizó una primera reunión con los miembros de la misma para que tuvieran dominio de qué era una Factoría de Software y sus ventajas desde el punto de vista personal, organizativo y productivo. Se explicó que las herramientas que se utilizan son las que determinan la línea de desarrollo de la factoría. Se les dio a conocer la nueva estructura de trabajo, roles y actividades que los miembros de la factoría debían desarrollar, incluyendo la familiarización con PSP. Por su gran importancia y utilidad se realizaron encuentros personales con los asesores de la factoría para explicar y ayudar directamente a cada desarrollador.

3.4.4 Repositorio

En un proyecto que ya ha comenzado a funcionar antes de probar la factoría ya deben existir componentes reutilizables, por lo que se debe hacer un catálogo para organizarlos y tener sus datos. En el proyecto todavía son pocos los componentes que se reutilizan por lo que el catálogo inicialmente se realizó en un documento Excel, donde se registraron los datos de los componentes reutilizables y utilizando las facilidades de las listas se pueden buscar por palabras clave, tipo o fecha de creación.

3.4.5 Personas

Para probar este paquete se asignaron los roles a nivel de factoría y de equipo, así como las tareas que cada uno debía cumplir de acuerdo a su rol de trabajo, según el sistema de conocimientos, habilidades y valores que deben tener para ello.

Los trabajadores que se destacaban por su trabajo y responsabilidad fueron designados a asumir el rol de líder de equipo o gerente de desarrollo y los que demostraron tener un conocimiento más profundo del trabajo con las herramientas se ubicaron como arquitectos. El gerente de la factoría se apoyó en los gerentes de desarrollo para controlar el trabajo de cada uno de sus miembros de equipo, lo cual contribuyó a aumentar la comunicación y entendimiento entre los desarrolladores.

Con relación al uso de PSP hubo una minoría de estudiantes que no estuvo de acuerdo por considerarlos como algo tedioso y sin objetivos prácticos. Por lo que se explicó con más detalles en que consistía el PSP y para facilitar su utilización se dio una plantilla de guía que contenía todas las herramientas de desarrollo de PSP. Se les especificó que la planificación era algo muy personal y que las plantillas no tenían un formato rígido, por lo que cada uno podía tener su propio estilo siempre y cuando le sirviera como disciplina para organizar su trabajo.

3.4.6 Bases tecnológicas

En este paquete se definieron los trabajadores que trabajaron con cada herramienta, así como las políticas para la seguridad de la información.

De los 22 desarrolladores, el equipo de actividades interactivas trabajó con la herramienta Macromedia Flash MX 2004. El resto trabajó con los lenguajes de programación HTML, JavaScript y CSS y la herramienta que utilizó para esto fue Macromedia Dreamweaver. El Microsoft Project fue utilizado por el gerente de la factoría para distribuir las actividades que serían chequeadas por los líderes en cada equipo de desarrollo. Además se establecieron políticas de seguridad de la información para las estaciones de trabajo, que fueron circuladas por los asesores de la factoría vía correo electrónico.

3.5 Reporte de los resultados

El reporte de los resultados se realizó aplicando encuestas a los integrantes del proyecto. Cuando se comenzó a probar la estrategia el proyecto se redujo de 35 desarrolladores a 22, de ellos solo 10 fueron encuestados.

El 70% manifestó tener conocimiento acerca del objetivo de aplicar el enfoque de factoría principalmente para la parte organizativa.

El 100% de los encuestados opinó que la nueva estructura organizativa era buena, de ellos el 67% afirmó que ayudaba a mejorar la comunicación entre los integrantes de los equipos y todos los tener

conocimiento del rol que desempeñaban, notando la diferencia producida en el proyecto después de probar la utilización de TSP propuesta por la estrategia con una mejor organización.

La utilización de la planificación que brinda PSP permitió controlar y planificar mejor el trabajo, además de brindar la posibilidad de ser evaluadas por sus jefes de equipo y por el líder del proyecto, permitiendo una mayor comunicación interpersonal. El 40% los encuestados demostró que con que la utilización de estas herramientas se pueden detectar errores y el 53% coincidió en que sirven de gran ayuda para planificar, evaluar y controlar sus actividades laborales. (Ver anexo 28)

3.6 Análisis de los riesgos de la transferencia

Algunos de los riesgos previstos se dieron durante la prueba piloto de la factoría, convirtiéndose en problemas para el funcionamiento correcto de la misma. Para estos riesgos se previó un plan de mitigación de forma tal que no todos se produjeron y para los que afectaron se tomaron las medidas propuestas en el plan de contingencia. A continuación se muestran todos los riesgos y la explicación de cada uno.

- R1: Este riesgo no afectó la estrategia ya que se coordinó una reunión inicial con el gerente de la factoría para explicarle todo lo referente al enfoque de Factoría de Software y lo que se quería hacer.
- R2 y R3: Estos riesgos no afectaron tanto el funcionamiento del proyecto porque los asesores trabajaron directamente con los trabajadores, y estuvieron atentos a cualquier duda que tuvieran en los temas de la factoría, buscando una mayor motivación y compromiso de trabajo.
- R4: Este riesgo no se pudo observar, ya que la estrategia no se implantó completamente, solo se probaron algunos paquetes.
- R5: Este riesgo no se dio porque se determinó con anterioridad los paquetes que serían probados.
- R6: El proyecto solo contó con 22 desarrolladores y su líder, por lo tanto para prevenir el riesgo se asignaron los roles en dependencia de la cantidad de trabajo y de las capacidades de cada desarrollador. Por ejemplo, la persona en el rol de gerente de la factoría asumió también el rol de administrador del repositorio.
- R7: Este riesgo si afectó la factoría porque no se impartieron todos los cursos previstos, incluso desde antes de probar la estrategia y los desarrolladores no adquirieron los conocimientos

necesarios. Sin embargo se trato de equilibrar en los equipos de trabajo los desarrolladores menos aventajados con los de mayor conocimiento en las tecnologías y herramientas a utilizar.

- R8: Para prever este riesgo se realizó una minuciosa revisión bibliográfica como parte de la investigación científica. Además los asesores consultaron a especialistas en el tema de factoría.

3.7 Conclusiones

La estrategia de transferencia del modelo de Factoría de Software aplicando inteligencia se comenzó a probar en el proyecto cuando éste llevaba tres meses de desarrollo, por lo que no se pudo comprobar en su totalidad.

En este capítulo se explicaron los resultados obtenidos después de haber probado algunos de los elementos propuestos en la estrategia de transferencia y se mostraron los riesgos previstos. Se llegó a la conclusión de que implementar en un proyecto una estrategia para transferir un modelo de Factoría de Software puede mejorar la organización de los recursos humanos durante la producción, ya que determinar una estructura organizativa correcta que funcione por equipos de trabajo, utilizar las herramientas idóneas tanto para el desarrollo de productos como para la organización y planificación personal, son elementos que demuestran el rendimiento y la efectividad de implantar una Factoría de Software.

CONCLUSIONES

Con la realización de este trabajo se adquirieron conocimientos importantes sobre las Factorías de Software que pueden ser aplicados a cualquier proyecto con características similares en busca de organizar y elevar la producción de software.

Se estudiaron los modelos más distintivos dentro de este nuevo enfoque para la producción. De ellos, se escogió el modelo aplicando inteligencia por considerarse el más completo de todos y por reunir las características más sobresalientes del resto de los modelos estudiados.

Esta investigación permitió cumplir el objetivo propuesto ya que sirvió como base para diseñar una estrategia de transferencia del modelo de Factoría de Software aplicando inteligencia para la línea de contenidos educativos. La estrategia no solo muestra en orden lógico los pasos necesarios para realizar la transferencia sino también define cada una de las entidades que propone el modelo específicamente para la elaboración de productos multimedias orientados a la Web.

Probar parte de la estrategia diseñada al proyecto fue más que una necesidad un reto. A pesar de no probarse en su totalidad y no haber brindado todos los resultados esperados sirvió de gran utilidad para la organización interna del trabajo; según las encuestas realizadas, la división del trabajo por roles fue satisfactoria para mejorar además de la organización, la fluidez de la comunicación entre los miembros de la factoría.

RECOMENDACIONES

El objetivo de este trabajo investigativo es diseñar la estrategia de transferencia de un modelo de Factoría de Software. Por lo que se considera necesario implantar completamente esta estrategia, ya que en este trabajo solo se prueban algunos elementos de la misma. Se recomienda:

1. Realizar un estudio de factibilidad económica que demuestre los beneficios económicos de aplicar la estrategia de transferencia del modelo.
2. Redefinir el modelo propuesto para otra línea de producción o proyecto productivo.
3. Implantar esta estrategia en proyectos similares desde sus inicios.

REFERENCIAS BIBLIOGRÁFICAS

- ACEVEDO, A., ZAMUDIO. *UN ANALISIS DE LA TRANSFERENCIA Y APROPIACION DEL CONOCIMIENTO EN LA INVESTIGACION DE UNIVERSIDADES COLOMBIANAS*. *Investigación y Desarrollo*, 2005. vol. 13: 128-157.
- ARENAS, M. I. G. *Curso XML*, 2007. [Disponible en: <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml#12> (12/11/2006)]
- ARNOLDO DÍAZ OLAVARRIETA, A. B. P. *Fábrica de software. Un modelo de negocio certificable basado en Estructura y Capacidades*. "Dime qué tienes, qué produces y te diré qué eres", 2006. [Disponible en: <http://www.certum.com/Publicaciones/FabSoft.pdf> (12/12/2006)]
- AZERTIA, G. *INFORMACIÓN GENERAL DE AZERTIA*, Corporación IBV, 2001. [Disponible en: <http://www.ibv.es/pages/empresas/azertia.html> (22/02/2007)]
- BASILI, V. R. C., G.; CANTONE. *A Reference Architecture for the Component Factory* *ACM Transaction on Software Engineering and Methodology* 1992.
- CANTONE, G. *Cantone, G. Software Factory: Modeling the Improvement*. 'Competitive Performance Through Advanced Technology'. *Third International Conference on (Conf. Publ. No. 359)*, 1992. 124 - 129. 127-129.
- CASAÑOLA, Y. T. *Propuesta de modelo de producción de software para la universidad de las ciencias informáticas*, Evento virtual. *Informática* 2007, 2007. [Disponible en: http://www.informaticahabana.com/evento_virtual/?q=node/160&ev=III%20Taller%20Internacional%20de%20Calidad%20en%20las%20TICs (12/01/2007)]
- CIENTEC. *CMMI: Mejorando Procesos en Forma Integrada.*, 2007. [Disponible en: <http://www.cientec.com/analisis/cmmi.asp> (12/03/2007)]
- CUSUMANO, M. A. *Software Factory: A Historical Interpretation*. 1989. pp. 23-30.
- EDCO. *Estándares de calidad organizacional*. , Fundación SES, 2002. [Disponible en: http://www.fundses.org.ar/archi/estandares_de_calidad.pdf (28/02/2007)]
- ESPAÑOLA, D. D. L. L. *Diccionario de la lengua española*. Madrid, 2005. p.
- FACTORY, V. S. *Web Corporativa Vector Software Factory*, Vector Software Factory, 2000. [Disponible en: <http://www.vectorsf.com/h/index2.html> (22/01/2007)]
- FERNANDES, A. A. T., DESCARTES DE SOUZA. *Fábrica de Software: Implementação e Gestão de Operações*. en. *ATLAS*, 2004.p.
- FERNSTROM, C. N., K. H; OHLSSON, L *Software Factory Principles, Architecture and Experiments.*, 1992, Vol. 9, No. 2: pp. 36-44.
- GARCÍA, J. L. U. *Teconologías Multimedia en el ámbito educativo*, 1998. [Disponible en: <http://www.sav.us.es/pixelbit/articulos/n10/n10art/art104.htm> (10/03/2007)]
- GARCÍA PELAYO Y GROSS, R. *Diccionario Pequeño Larousse Ilustrado*. 1992. p.
- HUMPHREY, W. S. *Introducción al Proceso de Software Personal*. Madrid, 2001a. p.
- . *Introduction to the Team Software Process*. 2001b. p.
- IBERMÁTICA. *GNSIS-Los métodos y técnicas a emplear*, 2007. [Disponible en: <http://www.ibermatica.com/ibermatica/integracion/gnsismetodos> (10/12/2006)]
- IVAN AAEN, P. B., LARS MATHIASSEN *The Software Factory: Contributions and Illusions.: Proceedings of the Twentieth Information System Research Seminar*. Scandinavia, Oslo, 1997.

- IVAR JACOBSON, G. B., JAMES RUMBAUCH. *El proceso unificado de desarrollo de software*. Madrid, 2000. p.
- JARITON, N. E. *Tecnologías*, 2007. [Disponible en: <http://www.alexandria.com.mx/tecnologias.php> (12/11/2006)]
- JAVERIANA, P. U. *Proyectos de transferencia tecnologica-CAP*, 2006. [Disponible en: <http://www.puj.edu.co/centro/cap/5891.htm> (22/03/2007)]
- JIMENEZ, S. V. *La Constitución venezolana, Volumen 2*. Ciudad Habana. Cuba, Universidad de las Ciencias Informáticas, 2005. p.
- 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.
- JULIO ARIEL HURTADO, C. B. *Hacia una Línea de Procesos Ágiles Agile SPsL Versión 1.0*. (FIET), F. D. I. E. Y. T., Universidad del Cauca, 2005.
- LAGUERRE, L. M. S. *SELECCIÓN DE ESTRATEGIAS*, 2007. [Disponible en: [http://www.gestiopolis.com/canales2/gerencia/1/selestra.htm\(13/06/2007\)](http://www.gestiopolis.com/canales2/gerencia/1/selestra.htm(13/06/2007))]
- MARTIN, I. S. A. *Marco Teórico sobre Planeación Estratégica*. Cuba, 2002.
- M.C. PAULK, B. C., M.B. CHRISSIS, C.V. WEBER. *Capability Maturity Model for Software, Version 1.1*, Software Engineering Institute, 1993.
- MICHELSON, B. M. *Plataforma Flash de Macromedia*, 2005. [Disponible en: http://www.adobe.com/es/platform/whitepapers/psgroup_flashplatform.pdf (18/11/2006)]
- MINTZBERG H. Y QUINN, J. B. Destreza en la estrategia. en: *El Proceso Estratégico*. HISPANOAMERICANA, E. P. H. Mexico, 1993.p.
- PATRICIO LETELIER, J. H. C., M^a CARMEN PENADÉS. *Métodologías Ágiles en el Desarrollo de Software*. DSIC -Universidad Politécnica de Valencia, 2004.
- PERALTA, M. D., E.; BRITOS, P. Y GARCÍA MARTÍNEZ, R. . *Evaluación asistida de CMMI-SW*. . (CAPIS), C. D. I. D. S. E. I. D. C. Argentina, Escuela de Postgrado. Instituto Tecnológico de Buenos Aires, 2004.
- PIEEP, P. *NORMAS SERIE ISO 9000 E ISO 14000 APLICADAS A EFICIENCIA ENERGÉTICA*, SECRETARIA DE ENERGIA. DIRECCION NACIONAL DE PROMOCION, 2003. [Disponible en: <http://energia.mecon.gov.ar/pdfs/direccion%20nacional%20promocion/ure/EE%20-%20NORMAS%20ISO%20SERIE%209000%20-%2014000.PDF>(15/02/2007)]
- PRESSMAN, R. S. *Ingeniería del software, un Enfoque Práctico*. Quinta edición. Madrid, 2002. 84-481-3214-3219 p.
- REDR. *Acciones REDR*, 2007. [Disponible en: <http://www.redr.es/acciones/index.php?cat=3&sec=20> (13/06/2007)]
- RODRÍGUEZ, T. P. D. N. *Multimedia Libro Electrónico*, Instituto Tecnológico de Celaya, 2005. [Disponible en: http://sisinfo.itc.mx/ITC-APIRGG/Materias/Mat3/MM_Unid1.php#Def (05/11/2006)]
- SANTOS, V. M. R. *Curso JavaScript.*, Universidad de Jaen, 2007. [Disponible en: http://geneura.ugr.es/~victor/cursillos/javascript/js_intro.html (10/11/2006)]
- SENSO, J. A. *Microsoft: amplia gama de productos para Internet.: Revista internacional científica y profesional*, 1996.

- TAYLOR, J. *Nuevas funciones y ventajas de Dreamweaver 8*, Macromedia, 2005. [Disponible en: http://www.adobe.com/es/devnet/dreamweaver/articles/dw8_newfeatures.html (14/11/2006)]
- YADENIS PIÑERO, Y. M., ABEL LORENTE, DERMIS BÁEZ. *Sistema Metodológico para el desarrollo de Software Educativo*. XVI Forum Nacional de Ciencia y Técnica, Ciudad de la Habana, Cuba, Ministerio de la Informática y las Comunicaciones, 2006. p.
- YANOSKY RÍOS, M. M. *Modelo funcional de la Factoría de Software de la UCI para la línea Carrefour*. Ciudad Habana, Cuba, Universidad de las Ciencias Informáticas, 2005. p.
- ZOZAYA, V. P. *Las redes y la transferencia de Conocimiento*, Portal Iberoamericano de Gestión Cultural, 2006. [Disponible en: <http://www.gestioncultural.org/gc/boletin/2006/bgc14-VZozaya.pdf> (08/02/2007)]

BIBLIOGRAFÍA

1. ALLIANCE, A. *Manifiesto for Agile Software Development*, 2001. [Disponible en: <http://www.agilemanifesto.org/> (2/03/2007)]
2. AMBROSO, J. C. Las fábricas de 'software' buscan especialización y costes laborales más bajos, 2005. [Disponible en: http://www.elpais.es/articulo/tecnologia/fabricas/software/buscan/especializacion/costes/laborales/bajos/elpcibtec/20050512elpcibtec_3/Tes/ (10/02/2007)]
3. BÁEZ, M. P. La industria de Software, una oportunidad para México, 2002. [Disponible en: <http://www.enterate.unam.mx/Articulos/2002/enero/software.htm> (16/01/2007)]
4. BATES, T. Como gestionar el cambio tecnológico. Estrategias para los responsables de centros universitarios. Primera Edición. Madrid, España, Editorial Gedisa, 2001. p.
5. CADENA, A. A. D. L. Inducción a ISO 9001:2000, Universidad de Guanajuato, 2007. [Disponible en: <http://www.siiia.ugto.mx/iso/documentos/redtutoria/Curso%20Inducci%C3%B3n.pdf> (22/02/2007)]
6. CATALDI, Z. L., FERNANDO; PESSAQ, RAÚL; GARCÍA-MARTÍNEZ, RAMÓN. Metodología extendida para la creación de software educativo desde una visión integradora. Revista Latinoamericana de Tecnología Educativa, 2004. Volumen 2.
7. CMMI. *Capability Maturity Model Integration*, 2002. [Disponible en: <http://www.sei.cmu.edu/cmmi/cmmi.html> (12/03/2007)]
8. CUEVA, C. F. A. J. M. eXtreme Programming (XP): un nuevo método de desarrollo de software. Novática, 2002.
9. EAM. The Software Factory, 2007. [Disponible en: <http://www.thesoftwarefactory.net/> (20/02/2007)]
10. GABRIELA DÍAZ, M.-A. P., MARÍA ANGÉLICA; GRIMMÁN, ANNA C; MENDOZA, LUIS E. Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica, Universidad Simón Bolívar (USB), 2003. [Disponible en: <http://www.academia-interactiva.com/ise.pdf> (5/03/2007)]
11. FOWLER, M. *The New Methodology*, 2003. [Disponible en: <http://www.willydev.net/descargas/articulos/general/MetologiaXP.PDF> (3/03/2007)]
12. GOMEZ, A. D. *Programa Internacional Eureka de Proyectos de I+D*, 2007. [Disponible en: www.redit.es/pdfs/El%20proyecto%20estrat%C3%A9gico%20FACTORY.ppt (22/02/2007)]
13. HERNÁNDEZ, S. C. R. A. El Paradigma Cuantitativo de la Investigación Científica. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2002. p. ISBN: 959-16-0343-6
14. INSTED. Sitio oficial del Departamento Nacional de Software Educativo, 2006. [Disponible en: <http://www.insted.rimed.cu/> (5/03/2007)]
15. LEE W, O. D. *Multimedia based instruction*, Massachussets, 2000. p.
16. MARTINTO, M. P. C. P. El diseño de la investigación científica. Importancia y requisitos del diseño. Ciudad de la Habana, Cuba, Universidad de las Ciencias Informáticas, 2006.
17. ---. El proceso de investigación científica. Estructura. Referencias bibliográficas y Normas. Ciudad Habana, Cuba, Universidad de las Ciencias Informáticas, 2006.

18. NÚÑEZ, D. M. E. C. Niveles de estrategia para la integración del repositorio nacional de objetos de aprendizaje, Sistema de Universidad Virtual, U de G, 2005. [Disponible en: http://www.cudi.edu.mx/primavera_2005/presentaciones/chan.pdf (04/02/2007)]
19. PARRA, H. R. D. *La evaluación de la transferencia de conocimiento en la relación de cooperación Universidad-Empresa: una visión desde el contexto de la sociedad del conocimiento*. 2003. 34-44 p. 1317-8822
20. ROMERO TENA, R. Reflexiones sobre el software educativo, 2007. [Disponible en: <http://tecnologiaedu.us.es/rromero/reflexiones.htm> (02/12/2006)]
21. ROSELLÓ, E. G. Una propuesta para la reutilización de componentes en el proceso de desarrollo de software educativo, 2002.
22. SANCHEZ, C. S. *Capitulo 4: javascript*, 2006. [Disponible en: <http://carlos-serrano-sanchez.blogspot.com/2006/11/capitulo-4-javascript.html> (10/11/2006)]
23. SAUER, S.-E. G. *MVC-Based Modeling Support for Embedded Real-Time Systems*, 1999.
24. SEI. *Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method definition Document* CMMI Integration Project. Software Engineering Institute, 2007. [Disponible en: <http://www.sei.cmu.edu/publications/documents/01.reports/01hb001.html>. (18/02/2007)]
25. VALDÉS PARDO, V. G. *Nuevas tecnologías de la información y la comunicación*. Villa Clara, Cuba, 2003. p.

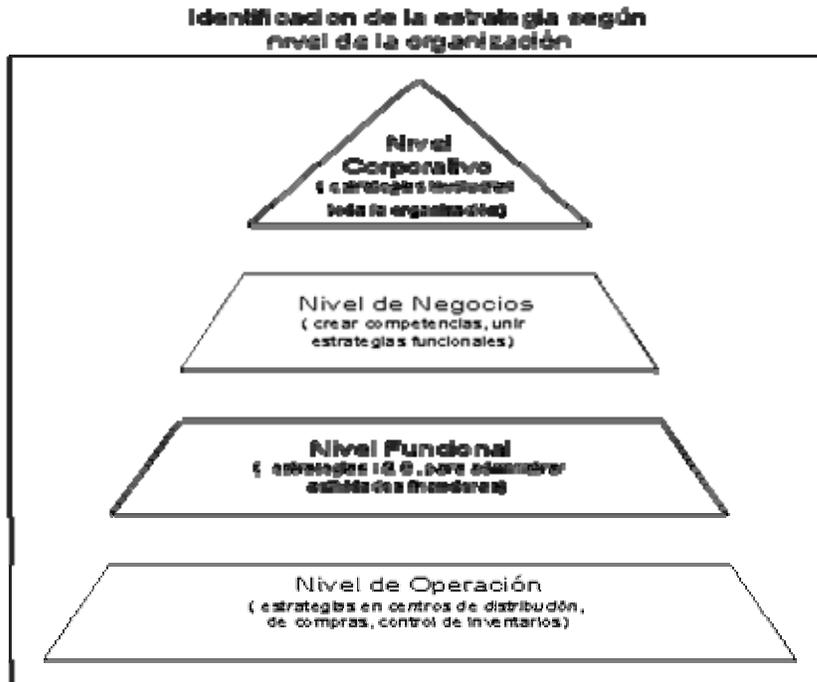
ANEXOS

Anexo 1: Comparación de los modelos de Factoría de Software

Aspectos	Modelos de Factoría de Software					
	ISO 9001 y CMM	Eureka	Clasificadorio	Basili	Replicable	Inteligencia
Proceso	Mejora continua del proceso de producción, pero sin definir el proceso	Desarrollo de software distribuido, guiado por reglas, no define su organización.	Especifica el proceso según el alcance	Especialización del área de producción de software y la de producción de componentes	Especialización del área de producción de software y la de producción de componentes y un modelo de calidad. Define un flujo de procesos pero no se guía por una metodología. No sirve para todo tipo de producto.	Define el proceso según una metodología de desarrollo. Vincular la mejora del proceso de producción a la gestión de proyectos y la orientación estratégica.
Clasificación de la Factoría	No menciona nada al respecto	No menciona nada al respecto	Clasifica la factoría según el alcance en: Proyectos Ampliada. Proyectos de Software. Proyectos Físicos. Programas.	No menciona nada al respecto	No menciona nada al respecto	Clasifica la factoría según el alcance de los flujos de trabajo dentro de una metodología de desarrollo de software.
Estándares de calidad	ISO 9001 y CMM, aunque este último se abolió en el 2005	Reglas	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	Sí, ISO 9001, CMMI, PSP y TSP
Gestión de proyectos	Define los roles, proceso y la calidad con la ISO 9001	Es responsabilidades de las reglas	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	Define los roles, procesos, alcance, costo, tiempo de desarrollo, plan de riesgos y la calidad.
Estructura organizacional	Los roles se agrupan en una entidad y utilizan PSP y TSP	Proceso modelado con reglas, tareas, actividades, y apoyada por	No menciona nada al respecto	No menciona nada al respecto	Los roles se establecen en función de los procesos	Las actividades y tareas de cada rol dependen de la

		herramientas automatizadas, pero no enuncia como organizarlo.				metodología, se apoya en herramientas automatizadas.
Bases tecnológicas	Define una entidad para el soporte técnico	Las herramientas soportan la automatización	No menciona nada al respecto	No menciona nada al respecto	Define una entidad para el soporte técnico y propone solo las herramientas iniciales.	
Reutilización de componentes	Para dar soporte al proceso.	La integración se realiza basado en reglas	No menciona nada al respecto	Establece los procesos para la gestión de la reutilización en el repositorio	Los utiliza para apoyar el proceso y los clasifica en activos del proceso o de infraestructura	Los utiliza para apoyar el proceso. Los clasifica en activos del proceso o de infraestructura y establece los procesos para la gestión de la reutilización en el repositorio
Gestión del conocimiento	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	No menciona nada al respecto	Plantea el uso de la inteligencia empresarial, vigilancia tecnológica, estudio de mercado, prospectiva, estudio de tendencias.

Anexo 2: Niveles de la Estrategia



Anexo 3: Diferencias entre las metodologías ágiles y las tradicionales

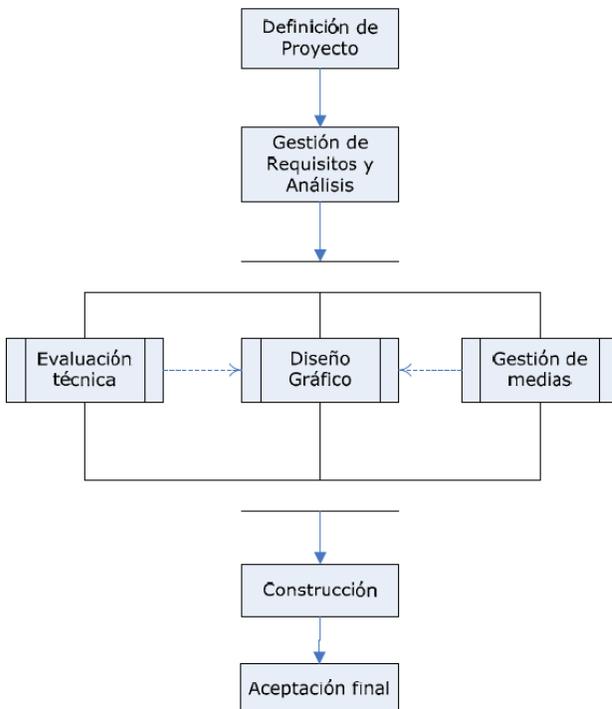
Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos

Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Anexo 4: Estructura de la Norma ISO 9001: 2000



Anexo 5: Flujo general de trabajo de la línea de Software Educativo



Anexo 6: Diseño de la encuesta realizada a los integrantes del proyecto antes de probar la estrategia

1. Que sabes acerca del proyecto CNTI - Contenidos Educativos:

2. Diga que deficiencias le encuentra al proyecto CNTI – Contenidos Educativos.

3. Diga la estructura organizacional del proyecto CNTI – Contenidos Educativos:

4. Marque con una cruz (X) su respuesta.

4.1 Conoces el rol que desempeñas en el proyecto. Si ___ No ___

En caso afirmativo diga cual: _____

4.2 Conoces algún proceso que se aplique al proyecto que organice el trabajo tanto personal como en equipo. Si ___ No ___

En caso afirmativo menciónelo (s): _____

4.3 Existe algún proceso que garantice la calidad de los productos elaborados por los integrantes del proyecto. Si ___ No ___
___ No se ___

4.4 Existe alguien encargado de supervisar la calidad de los contenidos.

Si ___ No ___ No se ___

4.5 Conoces el procedimiento que se sigue para elaborar una multimedia

Si ___ No ___

4.6 Diga si tienen establecidas políticas de seguridad de la información

Si ___ No ___ No se ___

En caso afirmativo menciónelas: _____

5. Cuantas personas integran cada equipo de trabajo: _____

6. Diga como es la comunicación entre los integrantes de cada equipo de trabajo

___ Buena ___ Mala ___ Regular ___ No se

7. Mencione (si conoce) la metodología que se utiliza en el proyecto para el desarrollo de las multimedia.

8. Diga si existe alguna parte donde se guarden los componentes programados para su posterior reutilización: _____

9. Mencione la herramienta que se utiliza para la gestión de proyecto:

10. Mencione la herramienta que se utiliza para la gestión de configuración:

11. Mencione los mecanismos que existen en el proyecto para garantizar el control de la calidad de los productos.

12. Mencione las tecnologías y herramientas necesarias para la creación y desarrollo de las multimedia.

Anexo 7: Diseño de la encuesta realizada a los integrantes del proyecto después de probar la estrategia

1. Que sabes acerca del proceso de la Factoría de Software:

2. Diga que deficiencias le encuentra al proyecto CNTI – Contenidos Educativos después de implantada la factoría.

3. Que opina acerca de la nueva estructura de trabajo.

4. Diga como ha sido la nueva distribución en equipos:

Buena Mala Regular

5. Cuantas personas integran cada equipo de trabajo: _____

6. Para qué te han servido las reuniones sistemáticas realizadas con el líder del proyecto.

7. Marque con una cruz (X) su respuesta.

7.1 Conoces el rol que desempeñas en el proyecto. Si No

En caso afirmativo diga cual: _____

8. Que opinas sobre la utilización de PSP

Me ha ayudado a planificar, evaluar y controlar mis actividades laborales.

Es una pérdida de tiempo y no sirve para nada.

Me permite detectar los errores y obtener productos con mayor calidad.

9. Mencione los mecanismos que existen en el proyecto para garantizar el control de la calidad de los productos.

10. En que te ha servido aplicar las nuevas técnicas para medir la calidad de los productos.

11. ¿Crees que sea necesario utilizar una metodología de ingeniería de software para desarrollar los productos multimedia?

Si No No se

12. Como ha sido la comunicación entre los integrantes de cada equipo de trabajo después de aplicar TSP.

Buena Mala Regular No se

13. Que crees de las plantillas y demás componentes reutilizables.

Disminuyó el tiempo de desarrollo. Disminuyó errores.

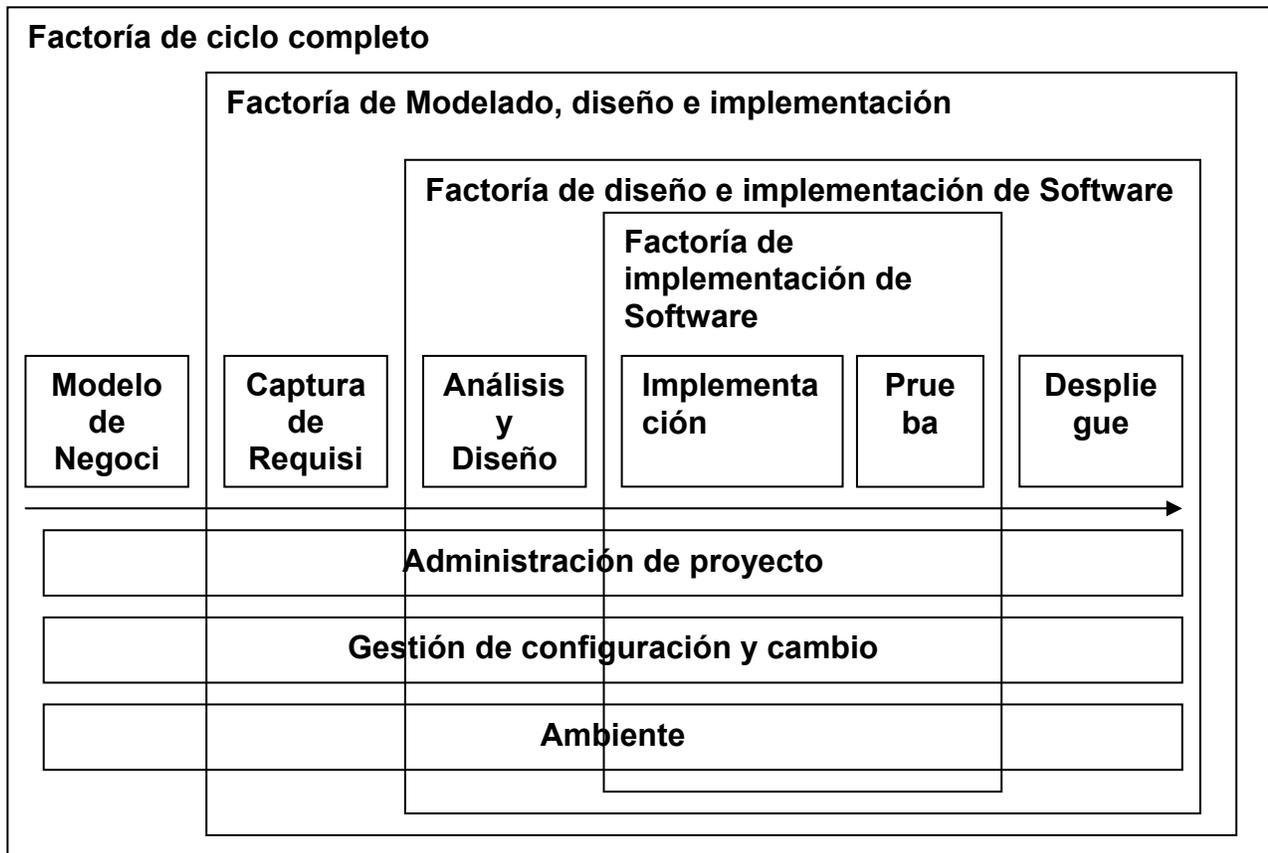
No sirvió para nada. Fue una pérdida de tiempo.

14. Diga que opina acerca del desenvolvimiento del proyecto después de implantar la Factoría de Software.

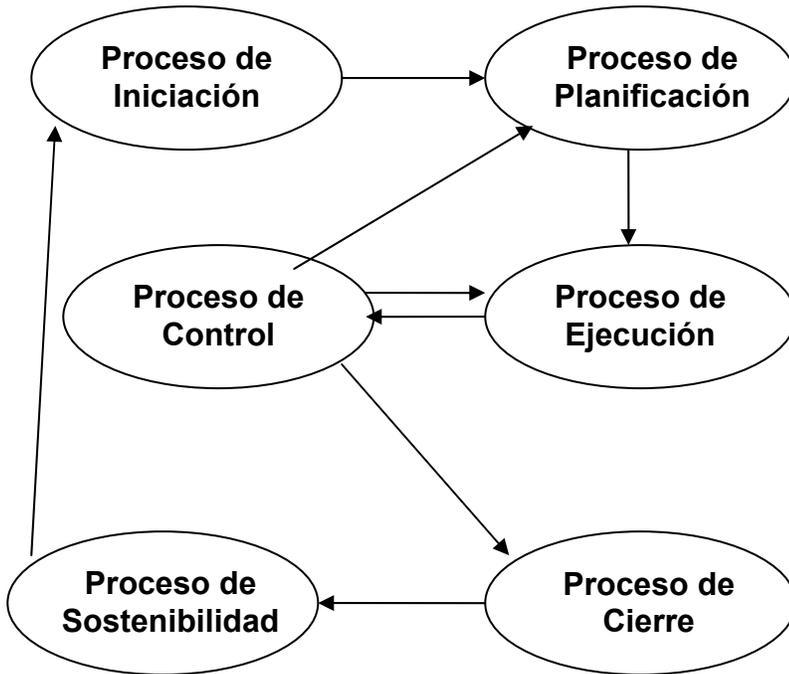
15. Si al implantar la factoría de software no se obtuvieron los resultados esperados fue por:

- ___ Errores o demoras en las entregas de diseño.
- ___ Falta de interés de los integrantes del proyecto.
- ___ Cambios constantes en los requisitos del cliente.
- ___ Mal diseño de la factoría.

Anexo 8: Clasificación de una Factoría de Software



Anexo 9: Flujo de procesos de gestión del proyecto



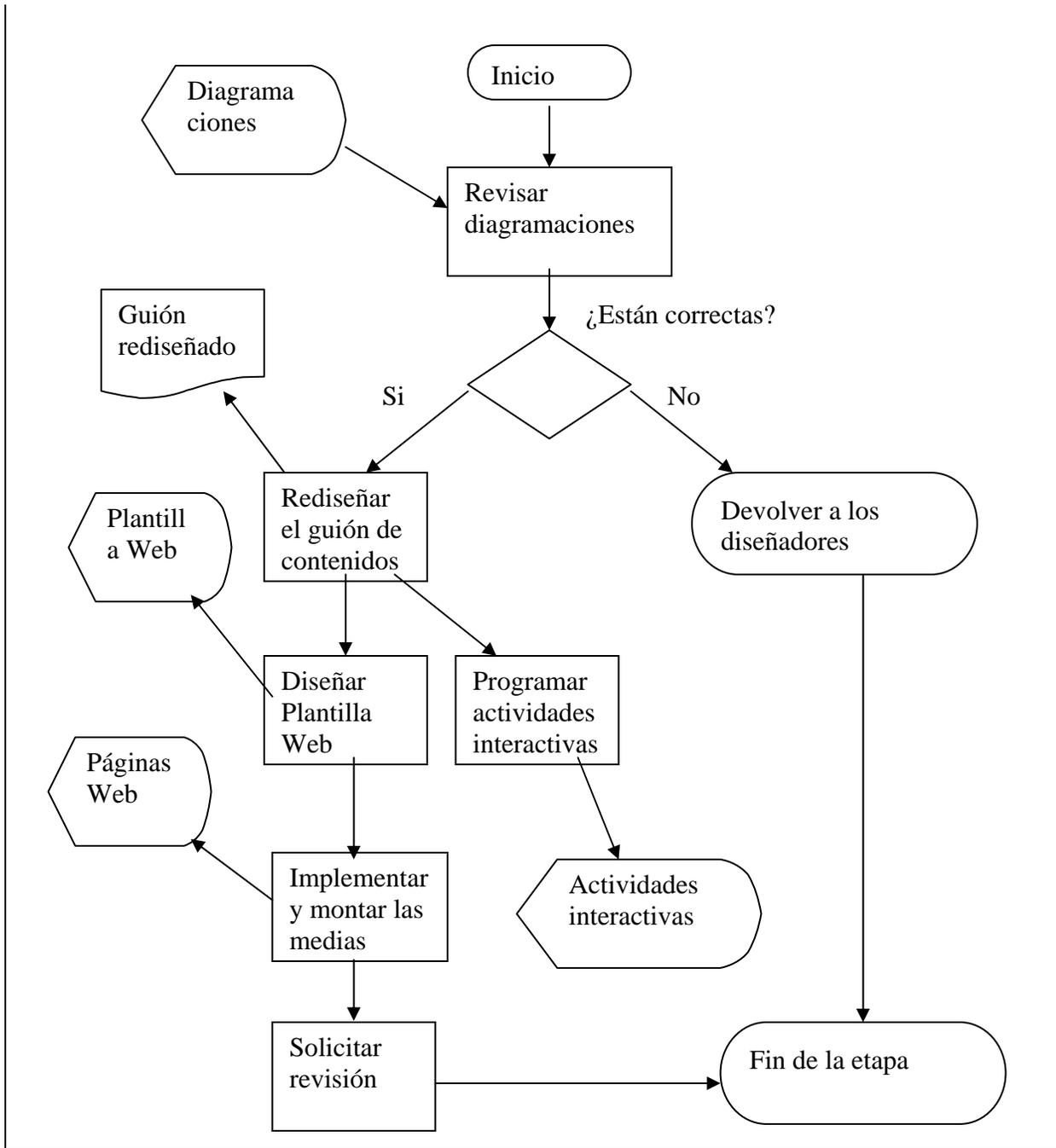
Anexo 10: Datos del equipo de trabajo

Datos del equipo de trabajo

Nombre y apellidos	Juan Pérez López		
Grupo	8301	Año que cursa	Tercero
Edificio	95	Apto	105
Teléfono	837 2945		
Correo electrónico	jperez@estudiantes.uci.cu		
No. Equipo	3	Rol	Programador
No. PC	1		

Anexo 11: Mapa del proceso de elaboración

Mapa de proceso (Etapa de Elaboración)



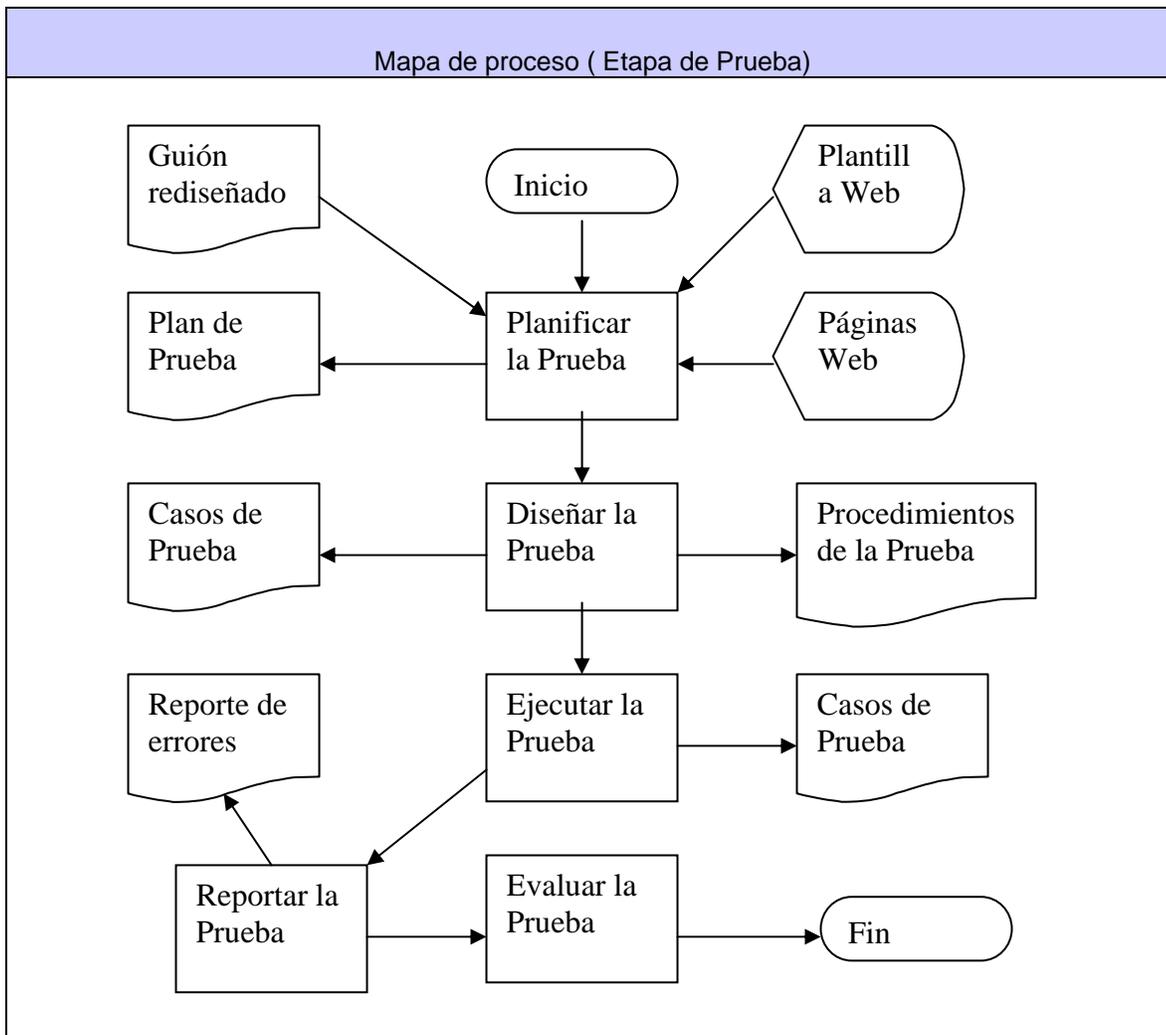
Anexo 12: Tabla descriptiva del flujo de procesos de elaboración

No.	Nombre de la actividad	Descripción	Artefactos de entrada	Tareas	Artefactos de salida	Rol	Bases tecnológicas
-----	------------------------	-------------	-----------------------	--------	----------------------	-----	--------------------

1	Rediseñar el guión de contenidos	El guión de contenidos es revisado y documentado en función de las diagramaciones.	Guión de contenidos. Diagramaciones.	1. Revisar el guión. 2. Revisar las diagramaciones. 3. Confeccionar el nuevo guión.	Guión rediseñado	Revisores	-
2	Diseño de plantillas	Se confecciona una plantilla Web estándar para todos los contenidos de un mismo portal.	Guión rediseñado	1. Analizar el guión rediseñado. 2. Definir hojas de estilo. (CSS) 3. Construir las plantillas	Plantilla Web del portal	Arquitectos	Dreamweaver
3	Agregar componentes reutilizables en caso necesario	Se identifican los componentes reutilizables para utilizar posteriormente en la programación.	Documento de Solicitud de componente al repositorio.	1. Identificar componentes reutilizables. 2. Solicitar componentes reutilizables al repositorio.	Componente reutilizable. Suministrado por: Arquitecto	Administrador del Repositorio	-
4	Programar actividades interactivas.	Se programan los juegos y actividades de interacción según el guión de contenidos.	Guión rediseñado	1. Analizar el guión rediseñado. 2. Buscar componentes reutilizables. 3. Programar las actividades.	Actividades interactivas	Programadores de Actividades Interactivas (Flasheros)	Flash
5	Planificar el montaje de las medias	Se le asigna a cada desarrollador las medias que cada uno debe montar en la plantilla.	Guión rediseñado. Medias del diseño. Plantilla Web.	1. Analizar el guión de contenidos. 2. Verificar las medias. 3. Distribuir los contenidos entre los desarrolladores	Plan de montaje Cronograma de trabajo	Gerente de la factoría y jefes de equipos de desarrollo	Project
6	Integrar medias a los contenidos.	Se montan las medias y actividades programadas en cada página del portal Web.	Guión rediseñado. Medias del diseño. Plantilla Web.	1. Verificar las medias. 2. Montar las medias en cada página.	Páginas del portal Web	Programadores de Interfaz Gráfica	Dreamweaver
7	Probar funcionalidades.	Hacer distintos tipos de pruebas para verificar la funcionalidad de los contenidos programados.	Guión rediseñado. Páginas Web.	1. Analizar el guión rediseñado. 2. Probar la navegabilidad y otras funcionalidades.	Páginas del portal Web	Programadores	Dreamweaver y Flash
8	Solicitar revisión.	Se solicita una revisión cuando se construye cada parte del producto.	Páginas del portal Web	1. Solicitar revisión de cada parte terminada del producto.	Páginas del portal Web	Programadores	-
9	Integrar	Se montan las	Páginas del	1. Analizar el	Portal Web	Arquitecto	Dreamweaver

contenidos al portal.	páginas y se integran todas al portal Web.	portal Web Guión de contenidos Diagramaciones	guión de contenidos y las diagramaciones. 2. Integrar las páginas al portal Web según el mapa de navegación.		
-----------------------	--	---	---	--	--

Anexo 13: Mapa de procesos de la prueba



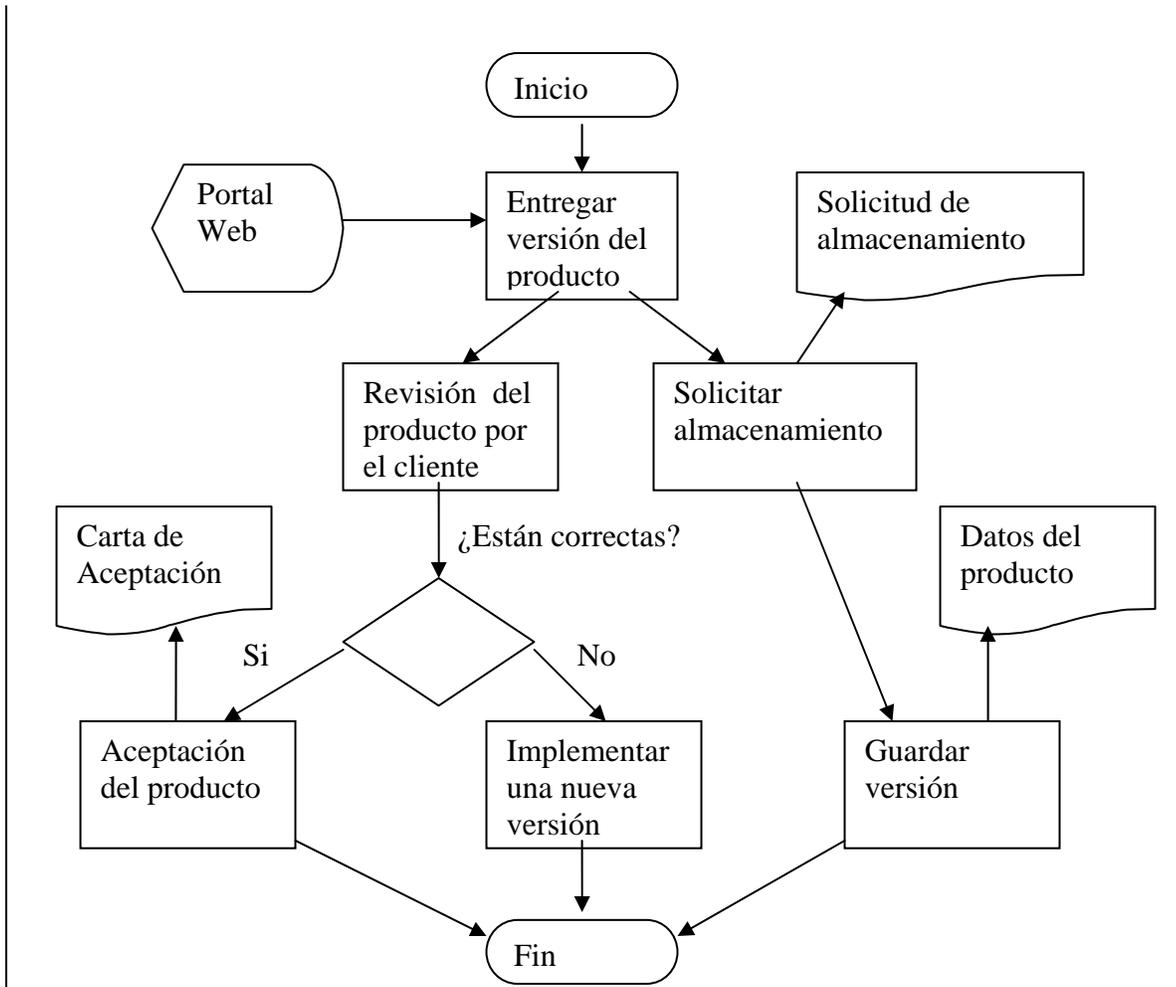
Anexo 14: Tabla descriptiva del flujo de procesos de la prueba

No.	Nombre de la actividad	Descripción	Artefactos de entrada	Tareas	Artefactos de salida	Rol	Bases tecnológica
-----	------------------------	-------------	-----------------------	--------	----------------------	-----	-------------------

							S
1	Planificar las pruebas	Se planifican las pruebas que se van a aplicar a cada producto, quienes lo harán y el tiempo de duración.	Guión rediseñado. Plantillas Web y productos a probar.	1. Definir una estrategia de prueba. 2. Definir los recursos necesarios. 3. Elaborar una planificación.	Plan de Prueba	Jefe de prueba	-
2	Diseñar las pruebas	Se diseñan los tipos de prueba que se van a hacer y los procedimientos que se seguirán.	Plan de Prueba Guión de contenidos	1. Definir las condiciones de prueba. 2. Elaborar los casos de prueba.	Casos de prueba y procedimientos de prueba.	Jefe de prueba	-
3	Ejecutar las pruebas	Se aplican las pruebas diseñadas a cada parte del portal Web.	Plan de prueba Casos de prueba y procedimientos de prueba Productos de portal Web	1. Detectar defectos. 2. Elaborar listado de errores.	Listado de errores	Probadores	-
4	Reportar errores	Se envía un listado con los errores encontrados en esa iteración.	Listado de errores	1. Enviar listado de errores	Reporte de errores	Jefe de prueba	-
5	Evaluar la prueba	Se comparan los resultados de las pruebas obtenidas con datos anteriores	Plan de prueba Casos de prueba y procedimientos de prueba Reporte de errores	1. Buscar datos de pruebas anteriores. 2. Analizar el cumplimiento del plan de prueba.	Evaluación de la prueba	Jefe de prueba	-

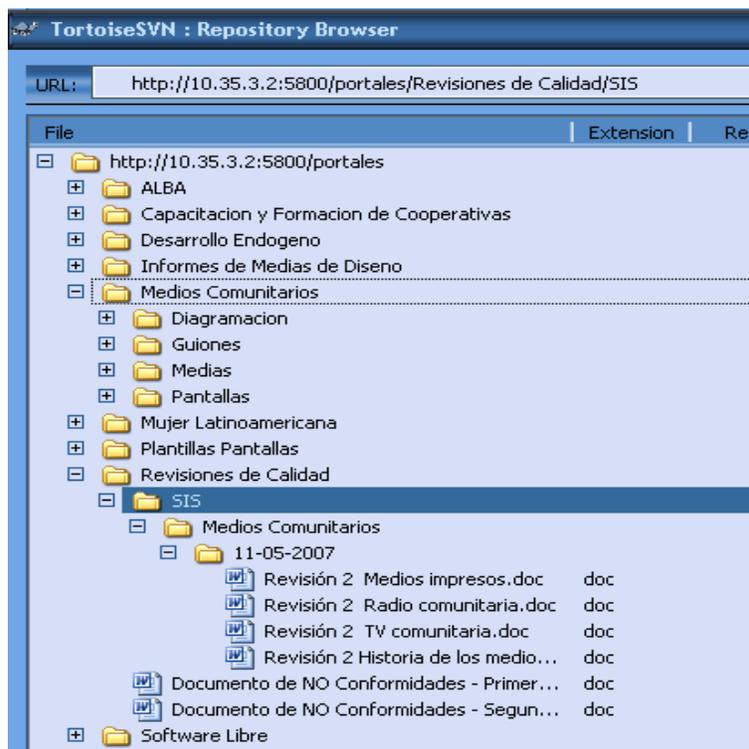
Anexo 15: Mapa del proceso de la entrega

Mapa de proceso (Etapa de Entrega)



Anexo 16: Tabla descriptiva del flujo de procesos de la entrega

No.	Nombre de la actividad	Descripción	Artefactos de entrada	Tareas	Artefactos de salida	Rol	Bases tecnológicas
1	Entregar versión del producto	Se entrega al cliente la versión terminada del portal Web.	Producto	1. Reunirse con el cliente. 2. Entregar el producto.	-	Gerente de la factoría	-
2	Proceso de aceptación	El cliente revisa y acepta el producto, sino pasa a otra fase de prueba.	Producto	1. Revisión del producto. 2. Firmar carta de aceptación.	Carta de aceptación del producto	Cliente	-
3	Almacenar versión del producto	Se almacena el producto construido en el repositorio de componentes.	Solicitud de almacenamiento Producto	1. Solicitar el almacenamiento del producto. 2. Insertar el producto.	-	Administrador del repositorio Gerente de la factoría	TortoiseSVN



Anexo 20: Conocimientos, habilidades y valores de los roles

Nombre del rol	Responsabilidades	Competencias		
		Habilidades	Sistema de conocimientos	Valores
Jefe de línea de desarrollo	Dirigir la línea de producción de software. Coordinar reuniones entre el gerente de la factoría y el cliente.	Ser positivo y cautivar la atención de los demás. Mantener la confianza en sí mismo y transmitirla a sus inferiores. Buen comunicador y negociador.	Conocimientos de dirección.	Educados, valientes y planificados.
Gerente de la factoría	Dirigir la factoría. Supervisar que el trabajo de todos los desarrolladores se termine dentro de lo previsto.	Líder por naturaleza. Ser positivo y cautivar la atención de los demás. Identificar las cuestiones claves y tomar objetivo decisiones. Buen comunicador, atento a las opiniones y eficiente ante los problemas.	Dominio del enfoque de factoría y del proceso de desarrollo de software.	Respetuoso. Preocupado por la superación de sus trabajadores.
Asesor de la factoría	Ayudar y asesorar al gerente de la factoría.	Buen comunicador y negociador.	Dominio del enfoque de factoría y del proceso de transferencia.	Educado, organizado y planificado.
Gerente soporte de	Apoyar a equipo en la determinación, la obtención, y el manejo de las herramientas	Ayudar al equipo en sus necesidades.	Experiencia con los instrumentos de apoyo y sistemas.	Organizado y actualizado en las tecnologías.

	Brindar ayuda administrativa.			
Planificador	Planifica todas las reuniones de la factoría y vela por el cumplimiento de todos los acuerdos tomados.	Planificar su trabajo. Coordinar el seguimiento de la planificación.	Conocimientos de planificación y de los procesos de desarrollo.	Organizado
Administrador del Repositorio	Administrar, configurar y mantener actualizado el repositorio de componentes.	Capacidad de administrar recursos	Dominio de la herramienta del repositorio.	Organizado
Gerente de desarrollo	Dirige cada equipo de desarrollo. Controla la calidad de los procesos y las actividades de los desarrolladores.	Líder por naturaleza. Escuchar las ideas de sus compañeros de equipo y comparar objetiva y lógicamente.	Experiencia en los procesos de desarrollo.	Disciplinado y autopreparado
Revisores	Revisan los guiones de contenidos y las diagramaciones para corregirlos y documentarlos.	Capacidad de retención y síntesis de la información.	Conocimiento en la confección de guiones de contenidos.	Organizados
Arquitectos	Elaborar las plantillas e integrar las páginas al portal Web. Identificar los componentes reutilizables.	Capaz de pensar en el software en un sentido amplio y centrarse en las partes críticas del desarrollo de un software.	Dominar la mayor cantidad de tecnologías de software y patrones de diseño.	Observador
Programadores de Interfaz Gráfica	Montan las medias a las páginas Web, las prueban y solicitan una revisión.	Asimilar rápidamente los conocimientos de programación y diseño visual.	Conocimiento de diseño y lenguajes para sitios Web.	Autopreparación
Programadores de Actividades Interactivas (Flasheros)	Programan las actividades interactivas, las prueban y solicitan una revisión.	Asimilar rápidamente los conocimientos de programación.	Conocimiento de flash y actionscript.	Autopreparación
Jefe de prueba	Planifica, diseña y supervisa la ejecución de las pruebas. Elabora un reporte de los errores encontrados.	Conocimiento del proceso de calidad de software. Asumir con naturalidad parte de la dirección de las pruebas.	Dominio de la ingeniería de software para pruebas.	Educado y organizado
Probadores	Ejecutan las pruebas y confeccionan un listado de errores encontrados.	Conocimiento del proceso de pruebas.	Dominio de la ingeniería de software para pruebas.	Organizados
Documentadores	Documentan los componentes y elaboran una plantilla de información de los componentes.	Buena redacción y ortografía	Dominio de los procesos.	Organizados
Gestor de información	Promueve el conocimiento dentro	Buen comunicador y negociador.	Dominio de los procesos y de las	Educados

interna	de la factoría. Gestiona cursos de capacitación a los desarrolladores.		metodologías de investigación.	
Gestor de información externa	Investiga el mundo de la tecnología exterior a la factoría.	Buen investigador y comunicador.	Dominio de los procesos y de las metodologías de investigación.	Educados y organizados
Gestor empresarial	Reúne la información de la gestión interna y externa y toma decisiones para la empresa.	Buena redacción y ortografía, excelente comunicador y negociador.	Dominio de los procesos y conocimientos de dirección empresarial.	Educados y organizados

Anexo 21: Diseño del Cuaderno del Ingeniero

En la portada se debe etiquetar el cuaderno con un número de cuaderno, debe tener nombre, número de teléfono, dirección de correo electrónico, fecha de comienzo de escritura de los datos y fecha del cierre. Por ejemplo:

<u>Cuaderno número: 1</u>	
Cuaderno de Ingeniería	
Proyecto CNTI-Contenidos Educativos	
Nombre del Ingeniero: <u>Pedro Pérez Pérez</u>	
Teléfono/ correo electrónico: <u>pperezp@estudiantes.uci.cu</u>	
Fecha de Apertura: <u>24/03/2007</u>	Fecha de Cierre:

Dentro del cuaderno se deben utilizar las dos primeras páginas como índice de contenidos y cada página debe ser numerada.

Ejemplo del cuaderno contenido de ingeniería

Página	Contenido del cuaderno de ingeniería Tema	1 Fechas
--------	--	-------------

Anexo 22: Diseño del Cuaderno de registro de tiempo

El cuaderno en su cabecera debe suministrar los datos referentes al nombre del trabajador de la factoría ya sea estudiante o profesor, la fecha de inicio, y el nombre o número de clase. En la Interrupción se refleja cualquier pérdida de tiempo, debido a interrupciones. El campo C (Completado) se rellena cuando se termina una tarea y la U (Unidades) es el número de unidades de una tarea acabada.

Fecha	Comienzo	Fin	Interrupcion	Δ Tpo	Actividad	Comentario	C	U
Semana1								
11/27/2005	8:00am	12:00pm		240	Descanso			
	1:00pm	5:00pm		300	Estudio GS	Leer cap 2,3 de PSP	x	2
	9:00pm	11:00pm		120	Estudio	Laboratorio		
10/28/2005	1:00pm	6:00pm		360	Estudio	Laboratorio		

Anexo 23: Diseño del Resumen semanal de actividades

Fecha
Inicio:

11/6/2005

Fecha
Fin:

11/12/2005

Tarea	Estudio	Clases					Total
D							0.00
L							0.00
M	360.00	360.00					720.00
Mi		120.00					120.00
J	360.00	120.00					480.00
V		270.00					270.00
S	180.00						180.00
Totales	900.00	870.00					1,770.00

Tiempos y Medias del Periodo

Numero de Semanas:

3

Resumen de las semanas anteriores							
Total	1,410.00	630.00					2,040.00
Med	705.00	315.00					1,020.00
Max	930.00	630.00					1,560.00
Min	480.00	630.00					1,110.00

Resumen incluyendo la ultima semana							
Total	2,310.00	1,500.00					3,810.00
Med	770.00	500.00					1,270.00
Max	930.00	870.00					1,800.00

Min	480.00	360.00				840.00
------------	--------	--------	--	--	--	--------

Anexo 24: Cuaderno de registro de defectos

Fecha: 11/14/2005

Programa: Prog 1 JS

Fecha	Numero	Tipo	Introducido	Eliminado	Tiempo de Correccion
11/14/2005	1	20	Codificacion	Compilacion	2
Descripcion	falta la llave de cierre del if-else				
	2	30	Codificacion	Pruebas	1
Descripcion	fichero no encontrado				
	3	50	Diseño	Compilacion	3
Descripcion	etiqueta de referencia a imagenes sin cerrar				
	4	20	Codificacion	Compilacion	1
Descripcion	punto y coma olvidado				
	5	20	Codificacion	Pruebas	3
Descripcion	no se especifica condicion de parada en el while				

Anexo 25: Descripción de las bases tecnológicas

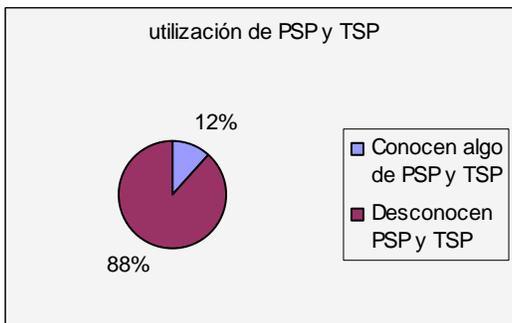
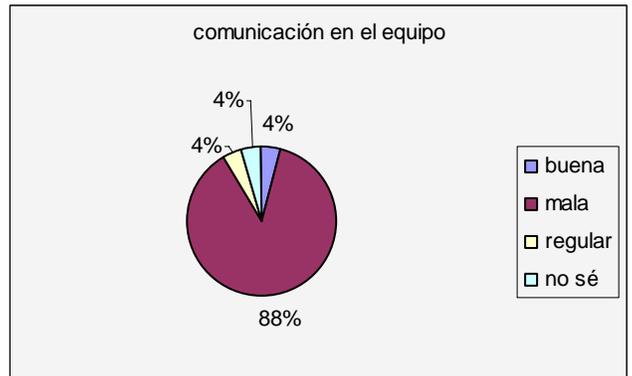
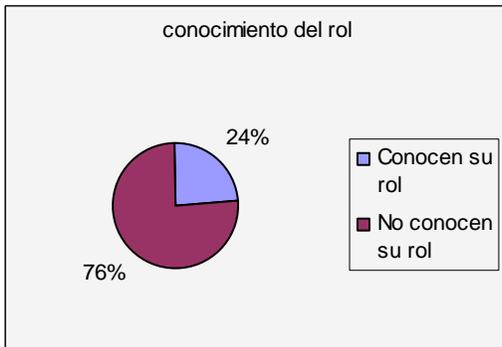
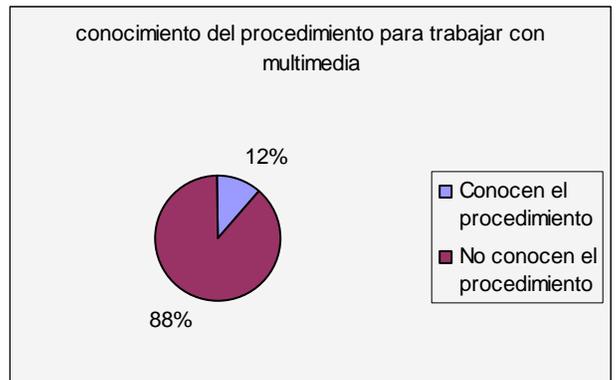
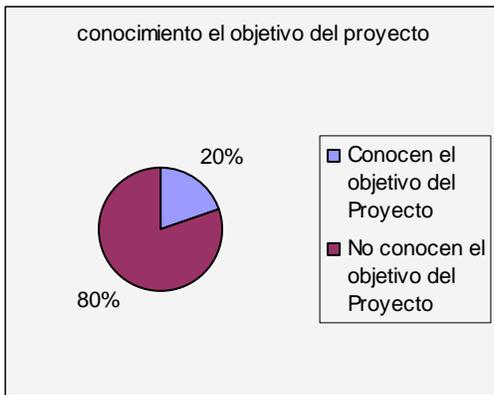
Nombre de la Bases Tecnológicas	Descripción
Dreamweaver / IDE	Diseño de aplicaciones Web rápidas.
Flash MX 2004	Diseño de actividades interactivas, ejemplo: los juegos.
Microsoft Project	Creación de planes de proyectos, asignación de tareas y recursos en un tiempo determinado.
TortoiseSVN	Utilizada para la configuración y control de versiones durante el proceso de desarrollo de software.

Anexo 26: Políticas de Seguridad de la Información (PSI)

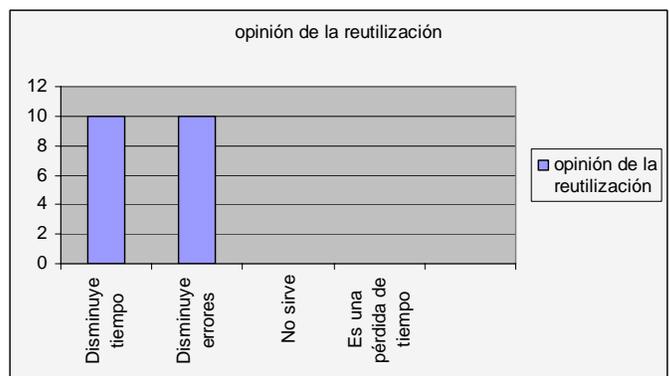
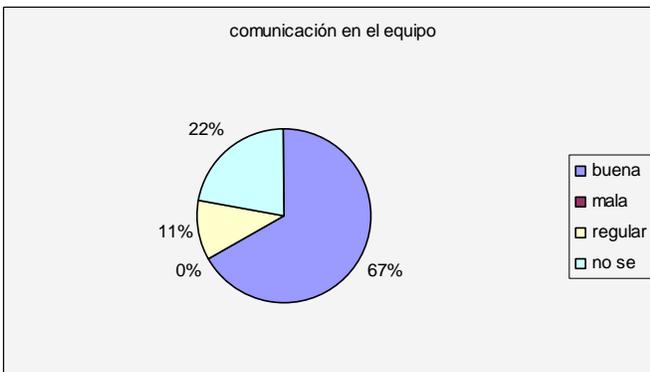
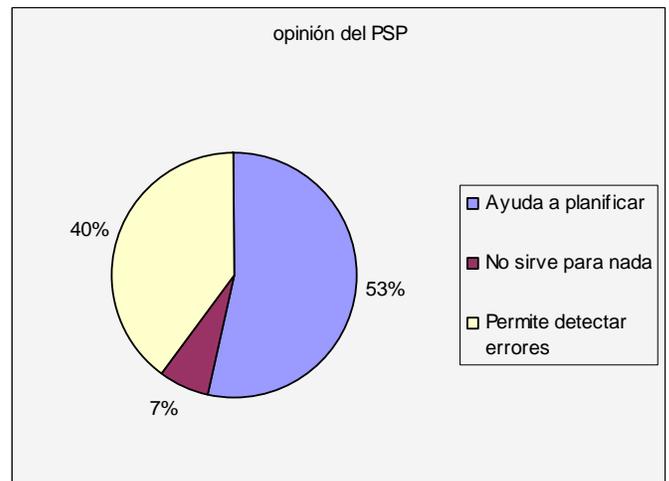
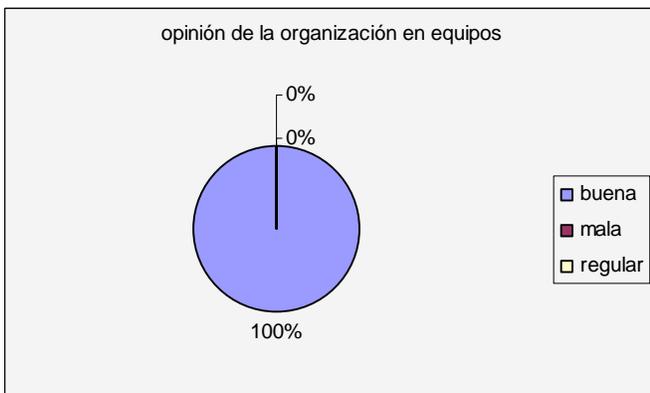
- Instalar un buen software antivirus y mantener actualizado su registro de virus diariamente.
- Instalar programas diseñados para prevenir el acceso a no autorizados (Firewall) de software o cualquier sistema seguro para controlar los puertos de su sistema.
- Usar Claves de Acceso que no estén asociadas a datos comunes del usuario.
- Cambiar de Clave de Acceso periódicamente.
- No ejecutar en las estaciones de trabajo ningún archivo contenido en un mensaje de correo no solicitado o enviado por un remitente desconocido.
- Borre constantemente las cookies, archivos temporales e historial, en la opción Herramientas, Opciones de Internet, de su navegador.
- Verificar frecuentemente los softwares instalados.
- Configurar los navegadores desactivando la ejecución automática de estos contenidos.

- No emplear los máximos privilegios en tareas, se debe configurar perfiles necesarios de acceso.
- Si el servidor no reconoce su nombre y clave de acceso o servicio de correo, podría ser que ya esté siendo utilizado por un intruso. A menos que haya un error en la configuración, la cual deberá ser verificada.
- Los productos en fase de desarrollo se tendría una copia local en varias estaciones de trabajo además de una en el servidor.
- En el traspaso de información al cliente es a través de un servidor FTP.

Anexo 27: Resultados de la encuesta inicial



Anexo 28: Resultados de la encuesta después de la estrategia



GLOSARIO DE TÉRMINOS Y SIGLAS

Términos

Análisis: Es un flujo de trabajo dentro de las fases de la ingeniería de software que se encarga de comprender con mayor exactitud los requisitos, refinarlos y estructurarlos. Proporciona una visión general del sistema de software.

Animación: Representación sucesiva de una secuencia de imágenes que produce la sensación de estar viendo imágenes en movimiento.

Arquitectura: Estructura organizativa de un sistema que incluye su descomposición en partes, su conectividad, mecanismos de interacción y principios de guía que proporciona información sobre el diseño del mismo.

Audio: Engloba todos los sonidos reproducidos por vibraciones de molécula, son percibidos por el oído humano y susceptible de medición. Entre ellos cuentan la música, voz y todos los demás ruidos.

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

Calidad de software: Grado con que el que un sistema, componente o proceso cumple con los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Clases: Es una abstracción de una entidad (objeto) concreta que existe en tiempo y espacio. Por lo que una clase es un conjunto de objetos que comparten una estructura y comportamiento común.

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

Componentes: Los componentes de software son los recursos desarrollados para un fin concreto y que puede formar solo o junto con otros un entorno funcional requerido por cualquier proceso predefinido.

Defectos: Se denomina defecto de un software a la falla o imperfección de un producto o proceso de software bajo sus condiciones normales de uso.

Diseño: Es un flujo de trabajo dentro de las fases de la ingeniería de software que se encarga de modelar el sistema y encontrar su forma para que soporte todos los requerimientos.

Diagramaciones: Presentación en flash de cómo debía quedar cada portal una vez elaborado o prototipo inicial del producto a elaborar.

Estándar: Un estándar define un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software.

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

Evento sistémico: Es un flujo de trabajo dentro de las fases de la ingeniería de software que se encarga.

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.

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

Hardware: Término que se utiliza para describir los artefactos físicos de una tecnología. Puede ser un equipo electrónico o informático. En la informática se denomina hardware al conjunto de materiales físicos que componen una computadora

Ingeniería de software: Es una tecnología multicapa en la que se identifican los métodos (que indican cómo construir técnicamente el software), el proceso (es el fundamento de la ingeniería de software) y las herramientas (soporte automático para el proceso y los métodos).

Implementación: Es un flujo de trabajo de la ingeniería de software donde se define cómo organizar los objetos en componentes de software. En esta etapa se construye el software y se obtiene un producto listo para probar y utilizar.

Implantación: Consiste en poner en producción el software obtenido. Esta fase de la ingeniería de software es la más extensa y la más sujeta a cambios en el ciclo de elaboración de un proyecto.

Metodología: Es un conjunto de métodos de investigación apropiados al quehacer de una ciencia determinada. Es la parte de un proceso de investigación que permite sistematizar los métodos y técnicas para realizar el proceso con eficiencia.

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

Multimedia: Es un sistema informático que utiliza varios medios de comunicación como textos, imágenes y sonidos para transmitir una información determinada.

Modelo de negocio: Describe los procesos de la empresa, identificando quiénes participan y las actividades que requieren automatización. Comprende la estructura y la dinámica de la organización en la cual se va a implantar el sistema, sus problemas actuales y evalúa el estado actual de la organización en la cual el sistema será explotado.

Norma: Especificación que reglamenta procesos y productos para garantizar la interoperabilidad. Son documentos con especificaciones técnicas de aplicación voluntaria.

Norma de calidad: Es un documento establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona, para un uso común y repetido, reglas, directrices o características para las actividades de calidad o resultados, con el fin de conseguir el grado óptimo de orden en el contexto de calidad.

Objeto: Un objeto es cualquier cosa real que se ofrece a la vista. En términos de informática un objeto es un tipo de dato abstracto definido por el programador de un software para almacenar datos y las funciones que operan sobre éstos. Los elementos de un objeto se les conoce como miembro y la funciones que operan sobre los datos se denominan métodos.

Paradigma: Un paradigma es un modelo o patrón en cualquier disciplina científica u otro contexto epistemológico.

Paradigma de programación: Representa un enfoque particular o filosofía para la construcción de un software. No existe uno mejor que otro sino que cada uno con sus ventajas y desventajas se adapta al proceso más apropiado.

Programación Extrema: Es una metodología de desarrollo de software ágil centrada en potenciar las relaciones interpersonales como clave para el desarrollo de software.

Proyecto: Es un esfuerzo temporal emprendido para crear un producto o servicio único.

Proyecto de software: Es un elemento organizativo a través del cual se gestiona el desarrollo del software. Obtiene un versión de un producto que contiene modelos, código fuente, documentación y un ejecutable.

Producto de software: Son los artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

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.

Prueba: Es un proceso que se desarrolla a lo largo del ciclo de vida de un software con la intención de encontrar errores previos a la entrega final. Es una actividad en la cual un sistema o componente es eficiente bajo unas condiciones o requerimientos específicos.

Requerimientos funcionales: Son las capacidades o funciones que el sistema de software debe cumplir.

Reutilización: Reutilizar es la acción de volver a utilizar los bienes o productos ya elaborados y probados. Puede venir propiciada por una mejora o restauración o sin modificarse, usarlo en la creación de un nuevo producto.

Repositorio: Es un sitio centralizado donde se almacena y mantiene información digital, como bases de datos o archivos informáticos.

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.

Tecnología: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

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

Siglas

CMM: Capability Maturity Model

CMMI: Capability Maturity Model Integration

CNTI: Centro Nacional de Tecnologías de la Información

CSS: Cascading Style Sheets

FDD: Feature Driven Development

HTML: HyperText Markup Language

I+D: Investigación y Desarrollo

IEEE: Institute of Electrical and Electronics Engineers

IE: Internet Explorer

ISO: International Organization for Standardization

LD: Lean Development

MVC: Model-View-Controller o Modelo Vista-Controlador

OMMMA-L: Object-oriented Modeling of MultiMedia Applications.

NS: Netscape

PSP: Personal Software Process o Proceso de Software Personal

PYME: Pequeñas y Medianas Empresas

RMM: Relationship Management Methodology

RUP: Rational Unified Process.

SCAMPI: Standard CMMI Appraisal Method for Process Improvement

SES: Sustentabilidad, Educación y Seguridad.

SEI: Software Engineering Institute

SGC: Sistema de Gestión de la Calidad

SIS-Copextel: Sistemas Informáticos de Software-Copextel

TIC: Tecnologías de la Información y las Comunicaciones.

TSP: Team Software Process o Proceso de Software en Equipo

UCI: Universidad de Ciencias Informáticas.

UML: Unified Modeling Language o Lenguaje Unificado de Modelado

WYSIWYG: What You See Is What You Get

XHTML: eXtensible Hyper Text Markup Language

XML: Extensible Markup Language

XP: Extreme Programming